# ICESat (GLAS) Science Processing Software Document Series

# Volume #
# GSAS Acceptance Test Procedures
# Version 2.0


**Jeffrey Lee/Raytheon ITSS**
**Observational Science Branch**
**Laboratory for Hydrospheric Processes**
**NASA/GSFC Wallops Flight Facility**
**Wallops Island, Virginia 23337**


**November 2001**

**ICESat Contacts:**


**H. Jay Zwally, ICESat Project Scientist**
***NASA Goddard Space Flight Center***
***Greenbelt, Maryland 20771***

**Bob E. Schutz, GLAS Science Team Leader**
***University of Texas Center for Space Research***
***Austin, Texas 78759-5321***

**David W. Hancock III, Science Software Development Leader**
***NASA/GSFC Wallops Flight Facility***
***Wallops Island, Virginia 23337***

# Foreword

This document contains the GLAS Science Algorithm Software (GSAS) Acceptance Test Procedures. This document is developed under the structure of the NASA STD-2100-91, a NASA standard defining a four-volume set of documents to cover an entire software life cycle. Under this standard a section of any volume may, if necessary, be rolled out to its own separate document. This document is a roll- out of the user guide within the Product Specification Volume.

The GEOSCIENCE LASER ALTIMETER SYSTEM (GLAS) is a part of the EOS program. This laser altimetry mission will be carried on the spacecraft designated EOS ICESat (Ice, Cloud and Land Elevation Satellite). The GLAS laser is a frequency-doubled, cavity-pumped, solid state Nd:YAG laser.

This document was prepared by the Observational Science Branch at NASA GSFC / WFF, Wallops Island, VA, in support of B. E. Schutz, GLAS Science Team Leader for the GLAS Investigation. This work was performed under the direction of David W. Hancock, III, who may be contacted at (757) 824-1238, hancock@osb.wff.nasa.gov (e-mail), or (757) 824-1036 (FAX).

The following GLAS Team members contributed to the creation of this document:

Raytheon / Suneel Bhardwaj

Raytheon / Anita Brenner

972 / David W. Hancock, III

Raytheon / Peggy Jester

# Table of Contents

# List of Figures

# List of Tables

Section 1

# Introduction

## 1.1 Identification of Document

This is the Acceptance Test Procedures document for the Version 2 delivery of the GLAS Science Algorithm Software (GSAS). The unique document identification number within the GLAS Ground Data System numbering scheme is TBD. Successive editions of this document will be uniquely identified by the cover and page date marks.

## 1.2 Scope of Document

The GLAS I-SIPS Data Processing System, shown in Figure 1-1, provides data processing and mission support for the Geoscience Laser Altimeter System (GLAS). I-SIPS is composed of two major software components - the GLAS Science Algorithm Software (GSAS) and the Scheduling and Data Management System (SDMS). GSAS processes raw satellite data and creates EOS Level 1A/B and 2 data products. SDMS provides for scheduling of processing and the ingest, staging, archiving and cataloging of associated data files. This document contains the Acceptance Test Procedures for the GSAS Version 1 delivery.

## 1.3 Purpose and Objectives of Document

The purpose of this document is to record the objectives, procedures, results and other technical information related to Acceptance Testing for the Version 2 delivery of GSAS.

## 1.4 Document Organization

This document's outline is assembled in a form similar to those presented in the NASA Software Engineering Program [Information Document 2.3a].

## 1.5 Document Status and Schedule

This is the initial version of the GSAS Acceptance Test Procedures for the V2 delivery.

## 1.6 Document Organization

This document's outline is assembled in a form similar to those presented in the NASA Software Engineering Program [Information Document 2.3a].

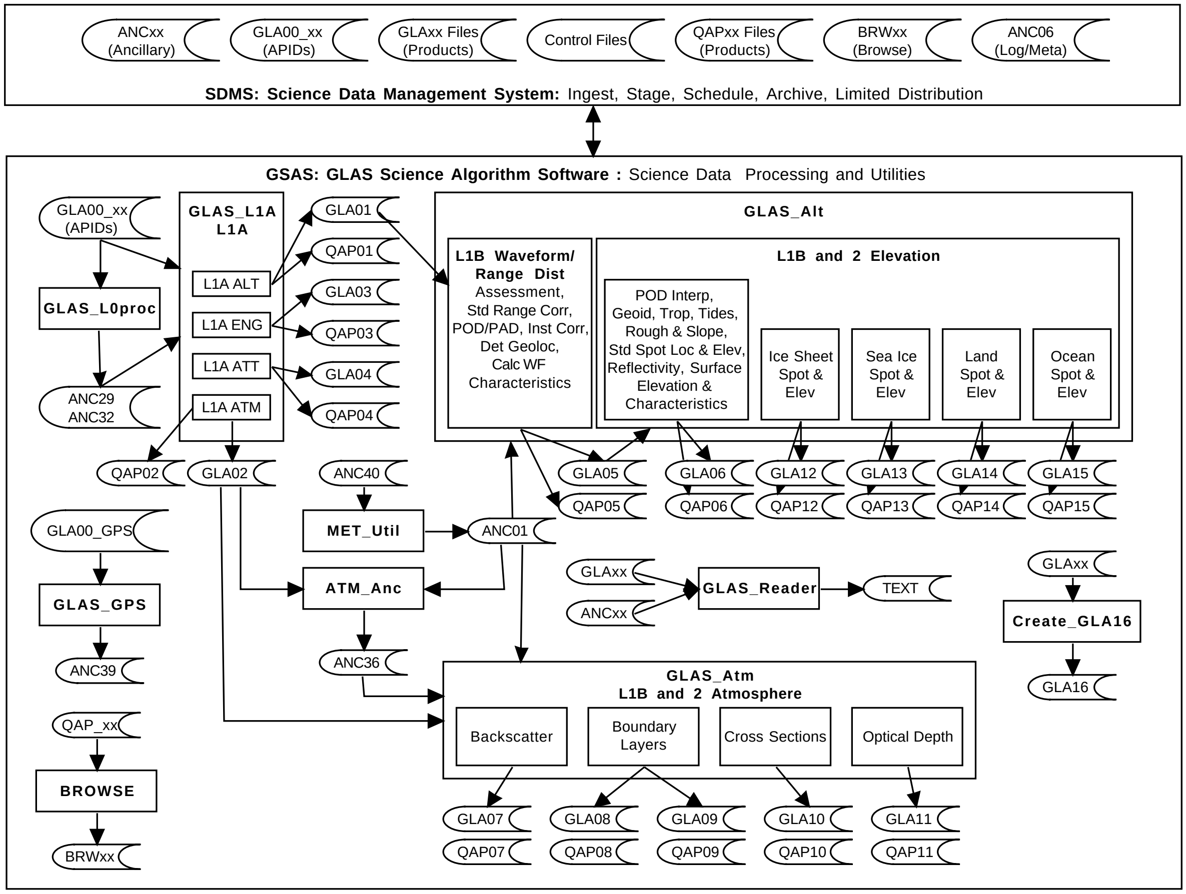


**Figure 1-1 I-SIPS Software Top-Level Decomposition**

## 1.7 Document Change History

| Document Name: GLAS Science Algorithm Software Acceptance Test Procedures | | |
|---|---|---|
| **Version Number** | **Date** | **Nature of Change** |
| Version 0 | July 1999 | Original Version. |
| Version 1 | November 2000 | Revised for V1 software. |
| Version 2 | November 2001 | Revised for V2 software. |

Section 2

# Related Documentation

## 2.1 Parent Documents

Parent documents are those external, higher-level documents that contribute information to the scope and content of this document. The following GLAS documents are parent to this document.

a) *GLAS Science Software Management Plan* (GLAS SSMP), Version 3.0, August 1998, NASA Goddard Space Flight Center, NASA / TM-1999-208641 / VER3 / VOL1.

The GLAS SSMP is the top-level Volume 1 (Management Plan Volume) document of the four volumes of NASA software engineering documentation [Applicable Reference 2.2c]. It dictates the creation and maintenance of the Product Specification Volume (Volume 2). This document is a roll out of the Product Specification Volume.

## 2.2 Applicable Documents

a) NASA Software Documentation Standard Software Engineering Program, NASA, July 29, 1991, NASA-STD-2100-91.

b) GLAS Science Algorithm Software Detailed Design Document, Version 2.0, November 2001, NASA Goddard Space Flight Center.

c) GLAS Science Algorithm Software User's Guide, Version 2.0, November 20001, NASA Goddard Space Flight Center.

d) GLAS Science Algorithm Software Version Description, Version 2.0, November 2001, NASA Goddard Space Flight Center.

e) GLAS ISIPS Operational Procedures Manual, TBD.

## 2.3 Information Documents

a) GLAS Level 0 Instrument Data Product Specification, Version 2.2, March 17, 1998, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2610.

b) GLAS Standard Data Products Specification - Level 1, Version 4.0, November 2001, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2621.

c) GLAS Standard Data Products Specification - Level 2, Version 4.0, November 2001, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2641.

d) GLAS Science Data Management Plan (GLAS SDMP), Version 4.0, June 1999, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DMP-1200.

Section 3

# Test Identification and Objective

## 3.1 Objective

Acceptance testing of GSAS is intended to verify that the “ready-to-be-delivered software” satisfies all the requirements as stated in or derived from the software requirements documentation. Version 2 acceptance testing will cover a subset of the requirements documented in the GLAS Science Software Requirements Document (GSSRD).

Acceptance testing will repeat tests carried out during Integration testing, using the same input data files and like control files that define the execution scenarios. A set of reference output files, which have been created and verified during Integration testing, will be used to validate like files created during Acceptance testing.

## 3.2 Test Identification

There will be five main tests that execute scenarios matching requirements defined in the GSSRD. The results will be compared with the results of the corresponding integration test cases. The following scenarios will be executed as required for Version 2:

**Table 3-1 Test Identification**

| Test | Description | Associated Requirements |
|---|---|---|
| 1 | Level 1A processing. Starting with GLA00, the software will create GLA01 and GLA02. Multiple granules of GLA01 and GLA02 will be created from a single GLA00. | GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30201, GSDP-30202, GSDP-30203, GSDP-30210, GSDP-30600, GSDP-30601, GSDP-30602, GSDP-31200, GSDP-31800 |
| 2 | Waveform processing. Starting with GLA01, the software will create GLA05. Multiple granules of GLA05 will be created from multiple granules of GLA01. | GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30205, GSDP-30210, GSDP-30700, GSDP-30701, GSDP-30703, GSDP-31200, GSDP-31800 |

**Table 3-1 Test Identification (Continued)**

| Test | Description | Associated Requirements |
|---|---|---|
| 3 | Atmosphere processing. The software will produce GLA07, 08, 09, 10, and 11 starting with GLA02 as input. A single granule (each) of GLA07-11 will be created from a single granule of GLA02. | GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30207, GSDP-30208, GSDP-30210, GSDP-30700, GSDP-30704, GSDP-30705, GSDP-30800, GSDP-30806, GSDP-30807, GSDP-31200, GSDP-31800 |
| 4 | Elevation processing. Starting with GLA05 product created in Test 2, the software will create GLA06, 12, 13, 14, and 15. | GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30206, GSDP-30209, GSDP-30210, GSDP-30700, GSDP-30702, GSDP-30703, GSDP-30800, GSDP-30801, GSDP-30802, GSDP-30803, GSDP-30804, GSDP-30805, GSDP-31200, GSDP-31800 |

## 3.3 Deviations from Integration Testing

None. The GSAS Acceptance Testing is a repetition of final GSAS V2.0 Integration Testing.

## 3.4 File Inventory

This section identifies the various files and directories used in GSAS Acceptance Testing. The runtime and source directories are fully described in the GSAS Version Description document.

The main acceptance test directory is located in $GLAS_HOME/test. The content of $GLAS_HOME/test is described in Table 3-2.

**Table 3-2 GLAS_HOME/test**

| Item | Description |
|---|---|
| anc | Directory containing input static ancillary files. Files required for each test are linked into the appropriate test subdirectory. |
| bin | Directory containing executables and shell scripts necessary to perform acceptance testing. Files required for each test are linked into the appropriate test subdirectory. |
| lib | Directory containing dynamic shared libraries. |
| setup_tests.sh | Shell script which links necessary files into each acceptance test directory. Will also verify the existence of the needed files in the source directories. (This is actually a link to the script of the same name within the bin directory). |

**Table 3-2 GLAS_HOME/test (Continued)**

| Item | Description |
|---|---|
| run_all.sh | Script which runs all GSAS acceptance tests. (This is actually a link to the script of the same name within the bin directory). |
| test1 | Directory for testing the GLAS_L0proc.(Test 1). |
| test2 | Directory for testing the GLAS_L1A .(Test 2). |
| test3 | Directory for testing the Utilities. (Test 3). |
| test4 | Directory for testing the GLAS_Atm (Test4). |
| test5 | Directory for testing the GLAS_Alt Waveforms. (Test 5). |
| test6 | Directory for testing the GLAS_Alt Elevation. (Test 6). |
| test_data. | Directory where test GLA, dynamic ANC, and reference files are stored. Files required for each test are linked into the appropriate test subdirectory. **Note that the output files from one test may be required as input files to another. In these cases, in order to preserve test independence, the input files are linked to the output reference files, not the newly-created output files**. |

The bin directory contains scripts needed during the acceptance testing process. This is different from the GLAS_HOME/bin directory, which contains the actual GSAS binaries.The scripts within $GLAS_HOME/test/bin are described in Table 3-3.

**Table 3-3 GLAS_HOME/test/bin**

| Item | Version | Description |
|---|---|---|
| GLAS_Alt | 2.0 | GLAS_Alt PGE. |
| GLAS_Atm | 2.0 | GLAS_Atm PGE. |
| GLAS_L0proc | 2.0 | GLAS_L0proc PGE. |
| GLAS_L1A | 2.0 | GLAS_L1A PGE. |
| GLAS_Reader | 2.0 | GLAS_Reader Utility. |
| SMDM_met_script | 2.0 | Script required by met_util. |
| add_ref.sh | 2.0 | Shell script which adds ‘.ref’ to provided file specification. |
| atm_anc | 2.0 | atm_anc utility. |
| clean_out.sh | 2.0 | This script will remove the outputs from the current run by parsing the requisite control file. This script is linked into each respective runtime test directory. |
| createGran_util | 2.0 | createGran_util Utility. |
| met_util | 2.0 | met_util Utility. |
| run_all.sh | 2.0 | Script which runs all GSAS acceptance tests. |

**Table 3-3 GLAS_HOME/test/bin (Continued)**

| Item | Version | Description |
|---|---|---|
| run_test?.sh | 2.0 | Script to automatically run each respective test. |
| setup_tests.sh | 2.0 | Shell script which links necessary files into each acceptance test directory. It is linked into the $GLAS_HOME/ test directory. |
| summarize_test.sh | 2.0 | This script will parse a control file and the header records of each output file to create a report containing the following information : file name, size in bytes, record length, number of header records, control start time, control stop time, header start time, header stop time, etc. |
| wgrib | n/a | Legacy executable required by met_util |

The test1-test6 directories are "runtime" directories which contain links to the binary, library, and data files needed in each respective acceptance test. The tester will perform each acceptance test within the respective test directory. The test directories are created by setup_tests.sh and, if removed, can be completely re-created by re-running the setup script.

The test_data contains sample product data, sample dynamic ancillary data, requisite control files, and reference data. The control subdirectory contains unique control files for each test. The input directory contains both input files and renamed links to reference files contained in the reference directory (for the case where the output of one test is required for input to another). The content of the test_data / input directory are listed in Table 3-4.Detailed descriptions of the GLA00 data are provided in the following subsection.

**Table 3-4 GLAS_HOME/test/test_data/input**

| Item | Description |
|---|---|
| anc01_001_2000010?_??0000_01.dat | Raw meteorological file data from NCEP (ftp.ncep.noaa.gov directory: /pub/data/grib/fnl) |
| anc01_001_2000010?_??0000_01_0?.dat | GLAS subsetted MET files. Links to the **reference** data created by met_util and verified by the science team during integration testing |
| anc08_001_20000101_000000_01_00.dat | Precision Orbit Data file, in house, based on Bob Schutz's 97 min orbit. |
| anc09_001_20000101_000000_01_00.dat | Precision Attitude Data file, in house from Bob Schutz. |
| anc12_000_00_00.dat | Digital Elevation Model - data. Provided by the science team. |
| anc12_000_00_01.dat | Digital Elevation Model - mask. Provided by the science team. |
| anc13_001_01_00.dat | Geoid. Provided by the science team. |

**Table 3-4 GLAS_HOME/test/test_data/input (Continued)**

| Item | Description |
|---|---|
| anc16_001_01_00.dat | Load Tide. Provided by the science team. |
| anc17_001_01_00.dat | Ocean tide. Provided by the science team. |
| anc20_002_20000101_000000_01_00.dat | Predicted orbit file. Provided by science team. |
| anc22_000_20000101_120000_01.dat | Reference orbit file. Provided by science team. |
| anc24_001_20000101_000000_01_00.dat | Rotation matrix. Provided by science team. |
| anc25_002_20000101_000000_01_00.dat | GPS-UTC time. Created by the development team. |
| anc26_000_20000101_120000_01.dat | |
| anc29_002_*.dat | APID Index files. Links to the **reference** data created by GLAS_L0proc and verified by the science team during integration testing. |
| anc32_002_*.dat | GPS Index files. Links to the **reference** data created by GLAS_L0proc and verified by the science team during integration testing. |
| anc33_002_20000101_000000_01_00.dat | |
| anc36_002_*.datl | Atmosphere calibration files**.** Links to the **reference** data created by atm_anc and verified by the science team during integration testing. |
| gla00_002_20000101_0?0000_01_??.dat | GLA00_01 APID data generated in-house by the development team. These data are fully described in the next section. |
| gla00_002_20000101_0?0000_02_??.dat | GLA00_02 APID data generated in-house by the development team. These data are fully described in the next section |
| gla00_002_20000101_0?0000_03_??.dat | GLA00_03 APID data generated in-house by the development team. These data are fully described in the next section |
| gla00_002_20000101_0?0000_04_??.dat | GLA00_04 APID data generated in-house by the development team. These data are fully described in the next section |
| gla01_002_*01_00.dat | GLA01 product files created from GLA00_01. Links to the **reference** data created by GLAS_L1A and verified by the science team during integration testing. |
| gla01_002_*02_00.dat | GLA01 product files created from GLA00_02. Links to the **reference** data created by GLAS_L1A and verified by the science team during integration testing. |
| gla01_002_*03_00.dat | GLA01 product files created from GLA00_03. Links to the **reference** data created by GLAS_L1A and verified by the science team during integration testing. |

**Table 3-4 GLAS_HOME/test/test_data/input (Continued)**

| Item | Description |
|---|---|
| gla01_002_*04_00.dat | GLA01 product files created from GLA00_04. Links to the **reference** data created by GLAS_L1A and verified by the science team during integration testing. |
| gla02_002_*01_00.dat | GLA02 product files created from GLA00_01. Links to the **reference** data created by GLAS_L1A and verified by the science team during integration testing. |
| gla02_002_*02_00.dat | GLA02 product files created from GLA00_02. Links to the **reference** data created by GLAS_L1A and verified by the science team during integration testing. |
| gla02_002_*03_00.dat | GLA02 product files created from GLA00_03. Links to the **reference** data created by GLAS_L1A and verified by the science team during integration testing. |
| gla02_002_*04_00.dat | GLA02 product files created from GLA00_04. Links to the **reference** data created by GLAS_L1A and verified by the science team during integration testing. |
| gla05_002_11_0001_*_00_00.dat | GLA02 product files created from GLA00_04. Links to the **reference** data created by GLAS_L0proc and verified by the science team during integration testing. |
| gla05_002_11_0001_*_00_00.dat | GLA05 product files created from GLA00_004. Links to the **reference** data created by GLAS_Alt and verified by the science team during integration testing. |
| reforbID*.dat | Reforbit data . Links to the **reference** data created by reforbit_Util and verified by the science team during integration testing. |

### 3.4.1 Detailed Description of GLA00_01 Input Data

The input GLA00 test data was created by the development team for the purpose of testing GSAS ground data processing. GLA00 test data format is based on document "GLAS Science Telemetry Packets, Version Rev B (3-Oct-00)".

The following data description was provided in the following document: ”*GLA00 Test Data Description*; Donghui Yi, Anita Brenner ITTS Raytheon GSFC; August 29, 2001”

**Summary:**

Some parameters in the test data have their sources. Some are reasonable guesses. And some are arbitrary set. Details will be described later. The atmospheric channel test data were provided by Steve Palm, Bill Hart, and Judd Welton (see Appendix A).

All parameters in GLAS science telemetry packet are defined in GLA00_prod_mod.f90 by Jeff Lee, Peggy Jester, and Greg Twigg.

**Satellite location**

The satellite location is calculated by using the eight-day repeat orbit data from Bob Schutz. The latitude, longitude and height values used in gla00 test data are linearly-interpolated values from the original latitude, longitude and height in the eight-day repeat orbit which is in 10 second interval. The first data point in gla00 test data start from the first point in eight-day repeat orbit file.

**Transmitted pulse**

The 48 gates transmitted pulse was a Gaussian pulse calculated using a sigma=1.699 ns. It peak value is at gate 25. Transmitted pulse is not changing for all records.

**DEM files**

DEM (1x1 degree) down loaded from NASA/GODDARD site: CoMITS

URL is: http://instra2.gsfc.nasa.gov:9001/comits/query/docs

This DEM is from Bob Schutz.

This DEM is used to determine surface elevation range (min and max) and surface type. Surface elevation for cloud measurement uses DEMmin-1km as surface elevation.

NOTE: There is a 1 km bias in this DEM. Actual elevation for DEMmin and DEMmax should be DEMmin-1km and DEMmax-1km.

DEM file from Suneel Bhardwaj in a fine scale (1x1 km) is used to determine elevation for a given location (lat,lon). This elevation is an input parameter used to calculate range between satellite and surface.

**Waveform files**

Waveform data are from SLA02 waveforms. First we separate land and ocean waveforms. We put all land waveforms (total 272986) in a file and all ocean waveforms (total 441546) in a file. Only GOOD WAVEFORMS (picked up by David Harding's group) are used. The noise level in the waveform and the standard deviation of noise came with the SLA02 data file. The SLA02 data files can be FTPed from: denali.gsfc.nasa.gov

SLA02 waveform has 150 gates, it will be extended to 200 or 544 by filling the empty gates with the mean noise level. When put into the telemetry data, the order of the waveforms will be reversed (latest in time goes first).

SLA02 waveforms are not related to the transmitted pulse used in test data. Its original gate size is 4 ns. When put into test data, 150 samples of sla02 waveform were put into 150 samples of test data (test data gate size is 1 ns).

200 gate waveforms are used for ocean and sea ice, and 544 gate waveforms are used for land and land ice. The type of surface is given in the 1x1 degree DEM.

No filter was applied to the waveform data.

No weight calculation was applied to the waveform data.

No gain was applied to the waveform data.

Therefore parameters describing the filter, weight, and gain information on the products are not consistent with the waveforms. All the parameters related to filter, weight, and gain are set to constant.

The 2-way travel time between satellite and earth surface is defined from transmitted waveform peak and the last peak of received waveform. The distance between satellite and earth surface is satellite height minus 1x1 km DEM height.

**GLA00 test data**

There are total 25 hours of data generated for each APID.

New data files will be generated if time pass any one of the 6-hour boundary (0:00, 6:00, 12:00, 18:00).

When all sla02 ocean or land waveforms were used up, we will reuse the waveforms from the beginning of the waveform files.

Files gla00_004_20010808_xxxx.dat are 25-hour data of APID 12, 13, and 19.

There are missing data and compressed data in the files.

The compressed records are in seconds

46001~46005 (p=2,q=1,n=100)

46021~46022 (p=4,q=1,n=120)

46501~46505 (p=1,q=2,n=130)

46521~46522 (p=1,q=4,n=100)

The missing data are in seconds

47101~47120 (APID 12 / 13, 19)

47101~47300 (APID 12 / 13, 19)

47501~47580 (APID 12 / 13)

47701~47710 (APID 12 / 13)

first three quarter second of 48201 (APID 12 / 13)

last two quarter second of 48210 (APID 12 / 13)

middle two quarter second of 48220 (APID 12 / 13)

last quarter second of 48230 (APID 12 / 13)

48231~48235 (APID 12 / 13)

first quarter second of 48236 (APID 12 / 13)

49001~49100 (APID 19)

**For APID 12, 13, 15, 17, and 19:**

Primary Header:

(3 bits) Ver#: set to 0

(1 bit) Type: set to 0

(1 bit) SEC Header Flag: set to 1

(11 bits) Application Process ID: 12, 13, 15, 17, and 19 for each related APID

(2 bit2) Segment Flags: set to 10

(14 bits) Source Sequence Count: incremented for each new packet created with same APID, range 0~16383

(16 bits) Packet Length: length of entire packet in bytes -7

Secondary Header:

(56 bits) Time Code (secondary_header_time): incremented for each new packet created with same APID)

initial value for secondary_header_time is:

start_MET_time-gd_Apid_offset(Index of APID)

second_header_time increase 1 second each time for APID 15, 17, and 19.

second_header_time increase 0.25 second each time for APID 12 and 13.

start_MET_time=1000 seconds in microseconds.

(8 bits) EDS Flag; set to 0

All shot counter values are between 1 and 200 and rollover to 1 after reach 200.

**APID 19: Ancillary Science Packet**

offset 14 shot counter: start from 41, increase by 40 (be careful, this is a 1 byte counter!).

offset 15: set to 31.

offset 16 shot counter: start from 1, increase by 40, always first of the 40 shots, -40 of

offset 14.

offset 18 Altimeter Dig. Range Window Rmin: same as offset 1100 Rmin but in ns.

offset 22 Altimeter Dig. Range Window Rmax: same as offset 1104 Rmax but in ns

offset 26 RMS Noise calculation start time offset: 6666 ns (1km in ns)

offset 30 Filter Selection Mask: set to 63.

offset 34 Shot Counter for PDL waveform: same as in offset 16.

offset 38 Post Delay Laser Pulse Response Start Address: set to 220000 ns.

offset 42 Sampled Post Delay Pulse Waveform: from transmitted waveform but shifted

offset 74 OTS Laser Pulse Response Start Address: set to 230000 ns

offset 78, 114, 150, 186:

Shot Counter for OTS#1: same as in offset 16

Shot Counter for OTS#2: Shot Counter for OTS#1+1

Shot Counter for OTS#3: Shot Counter for OTS#1+2

Shot Counter for OTS#4: Shot Counter for OTS#1+3

offset 82, 118, 154, 190:

Sampled OTS Pulse Waveform #1, 2, 3, and 4: from transmitted waveform but shifted.

offset 222 Location of transmit pulse search window (start): 195000 ns

offset 226 Number of No Threshold Crossing Shots for Error Condition: set to 1?

(? means we do not know what value should be set and just put a number there. we will see more ? later on.)

offset 231 Surface Echo Type: values come from 1x1 degree DEM. There are four values

(0: ocean & no ice, 1: land & no ice, 2: ocean & ice, and 3: land & ice).

offset 1140 is the same as offset 231.

offset 232 # elements averaged at ratio 'p' for frame: 200 for ocean, ocean & ice

waveform, 544 for land and land & ice waveform.

offset 234 Value of 'p' used for frame: set to 1.

There is no gate compression for this entire test data set. So offset 236 # elements averaged at ratio 'q' for frame and offset 238 Value of 'q' used for frame are both set to 0.

offset 250-342 Filter Weight Parameters:

Filter weight parameters are from "GLAS FLIGHT SCIENCE DATA SELECTION ALGORITHMS FOR THE ALTIMETER," Version 4.13, Appendix C. C0, C1, C2, C3 are the same for all six filters. C0=3.0, C1=-0.01, C2=-100.0, and C3=15.0. No filtering has been applied to the waveform data in the test data!

offset 346-414 Background Noise Coefficient:

Background noise coefficients are from Sam Orhing. A1, A2, and A3 are the same for all the filters. A1=4.5, A2=0, and A3=0. In test data, Background noise and Standard deviation of the noise are directly from SLA02 data and not been calculated using these coefficients.

offset 419 and 420 are set to 0?.

offset 421 Return Gain Value: set to 10.

offset 422-477 Auto Gain Calculation Parameters:

Some of the Auto Gain Calculation Parameters are from Sam Orhing. Some are

arbitrarily set. No auto gain calculation has been performed in creating test data!

A1=-0.7257, A2=0.7257, A3=0, A4=0, B1=0.0512, B2=0.8708, B3=0, B4=0, C0=0?, C1=0, Vref=1.5, Zmin=1?, Zmax=1?, Vmin=1?, Ginit=1?, Gmin=0.1,

Gmax=117.0. Please note that Vmin, Ginit, Gmin, and Gmax are 1 byte variables and this need to be checked!!

offset 478 Tolerance for Coincidence of Filters: set to 1?

offset 482-506 Range Window Dump Offsets: set to 40 for all filters.

offset 506 Surface (Pulse) Return Threshold Values for All Filters:

set to 1,2,3,4,5,6,7,8? for each byte.

offset 514 Fir Filter Coefficient: This parameter is in Little Endian format!

set to 3,5,7,9,11,13,15,17? for each byte.

offset 522 Filter Weight Min Std Deviation: set to 0.01?

offset 526-546 Filter Noise Minimum threshold: set to 2? for all filters.

offset 572-578: set to 0, not been used anymore.

offset 582-586:

SPCM Gate Delay and Background #1 Delay: both are 2.8 msec in 128 ns

Background #2 Delay: 4.3 msec in 128 ns

(ICESAT GLAS Flight Software, Photon Counter Software Specification and User's Guide, Version 1.0, Page18)

Range Gate Delay: Hsat-[Hmin-Hoffmin]+PCrangeBias in 128 ns

offset 590 SPCM status: set to 0?

offset 594 A / D output and CD Amplifier Attenuation (gain) setting: set to 1 and 1.

offset 596 Background #1 Delay: same as in offset 582

offset 598 Background #2 and Range Gate Delay: same as in offset 586.

offset 602 Detector status: set to 0?

offset 606 Shot Counter for start of frame: same as offset 16.

offset 608-1083 Shot Counter, Fire Acknowledge Time, and Fire Command Time:

There are 40 shots in a second, each shot has the above three parameters.

The first shot counter has the same value as in offset 16, for each shot it increasing by 1.

The first Fire Acknowledge Time start from 3000 second (46875000000 in 64 ns), for each shot it increasing by 0.025 sec (390625 in 64 ns).

Fire Command Time is 195 micro sec (3047 in 64 ns) before Fire Acknowledge time.

offset 1088-1092

Latitude, Longitude, and Height: Calculated from Dr. Bob Schutz's 8-day repeat orbit. Since the 8-day repeat orbit data has a interval of 10 seconds between data samples, the Latitude, Longitude, and Height are linearly-interpolated. The starting point for APID 19 is the same as the starting point in 8-day repeat orbit.

offset 1096 Rsat:

Rsat2 = r2 = x2+y2+z2. x, y, and z are the geocentric earth-fixed rectangular coordinates of spacecraft. x, y, and z are calculated from lat (f), lon (l), and height (H) in eight-day repeat orbit file.

x = r*cos(f')*cos(l)

y = r*cos(f')*sin(l)

z = r*sin(f')

f' is geocentric latitude of satellite.

f' = f's+arcsin[H / r*sin((f-f's)]

r = sqrt(H2+R's2+2HR's*cos(f-f's))

R's = ae(1-f) / sqrt((1-f)2cos2f's +sin2f's)

f's = arctan[(1-f2)sin(f) / cos(f)]

ae = 6378136.3 and f = 1 / 298.257 come with the eight-day orbit file.

(Anita: This is from a reference you given to me a few years a ago)

offset 1100 Rmin and 1104 Rmax:

Rmin = Hsat-[Hmax+Hoffmax]+Rbmin

Rmax = Hsat-[Hmin-Hoffmin]+Rbmax

offset 1108 Wmin: set at 2 km.

offset 1112 Wmax: set at 11 km.

offset 1116 Hoffmin: set at 1.5 km.

offset 1120 Hoffmax: set at -0.5 km.

offset 1124 Rbmin: set at 0.

offset 1128 Rbmax: set at 0.

offset 1132 PC Range Bias: set at -41 km.

offset 1136 CD Range Bias: set at -41 km.

offset 1140 Surface type: same as offset 231.

offset 1141 Position data valid flag: set to 0.

offset 1142 Spacecraft time & position packet data:

> bvtcw: start from 4008.5 sec in micro sec. Increase by 1 second for each record.
>
> x, y, and z see offset 1096 Rsat.
>
> GPS receiver time: start from 2000 sec in sec. Increase by 10 seconds every 10 records.
>
> bvtcw @ 0.1 Hz: start at 4000 sec in sec. Increase by 10 seconds every 10 records.

offset 1182 Shot Count for 1553 Spacecraft Position and command packet: offset 16 value+4, values between 1~200.

offset 1184 GLAS MET for 1553 Spacecraft position and command packet:

> start at 1000.1 sec in micro sec. In crease by 1 sec for each record.

offset 1192 DEM minimum byte & offset 1193 DEM maximum byte:

> DEM min and DEM max in the last 7 bits of each byte. Unit 125 m.
>
> DEM minimum first bit: 0-ocean, 1-land.
>
> DEM maximum first bit: 0-no ice, 1-ice.

offset 1194 Range data source: set to 1?

offset 1195 GPS 10 sec Pulse 40 bit count value: start at 2998.89. Increase 10 sec every 10 sec in 64 ns.

offset 1200 GLAS MET for GPS 0.1 Hz pulse: start at 998.89. Increase 10 sec every 10 sec in micro sec.

offset 1216 to offset 1244: total 16 parameters, values set from 1 to 16, just to distinguish them.

offset 1248 Dual pin A: set to 5.

offset 1288 Dual Pin B: set to 5.

offset 1328 532 Energy: set to 100.

**APID 12 & 13 Altimeter Digitizer - Large & Small Science Packet**

The difference between Large & Small Science Packet are:

SLA02 land waveforms were used in Large Science Packet.

SLA02 ocean waveforms were used in Small Science Packet.

The 150 gate SLA02 waveforms were extended to 544 for Large Science Packet waveforms, and extended to 200 for Small Science Packet.

offset 16 Shot Counter: The first record start from 1, continues count with each laser shot.values from 1 to 200, after 200, goes back to 1. Dose not distinguish ocean or land. For example, if Shot Counter is 40 when land ends and ocean starts, Shot Counter will be 41 for the ocean record.

offset 20 Transmit Pulse waveforms: 48 samples, values between 0 and 255. Keep same for the entire test data.

offset 68 Transmit pulse waveform peak time: 200000+int(300*sin(i+j))*4+25 in ns. i

and j are indexes in the program, i: 1~172800, j: 1~40.

offset 72 Transmit Pulse Waveform Peak Threshold Flag: set to 0.

offset 76 Starting address of transmit pulse sample: 200000+int(300*sin(i+j))*4 in ns.

i and j are indexes in the program, i: 1~172800, j: 1~40.

offset 80 Ending address of Range Response Surface Echo Dump:

The travel time between transmitted pulse peak and the last peak (in time) of the received pulse equals to the travel time between satellite and surface of the ground.

Ending address of Range Response Surface Echo =

Transmit pulse waveform peak time +

The travel time between satellite and surface of the ground +

150 (sla02 waveform total gate) -

LastPeakLocation (within 150 sla02 waveform gates)

offset 84 Last Threshold Crossing time:

Last Threshold Crossing time =

Transmit pulse waveform peak time +

The travel time between satellite and surface of the ground +

LastThresholdLocation (within 150 sla02 waveform gates)-

LastPeakLocation (last peak location of sla02 waveform with in 150 gates) offset 88 Next to Last Threshold Crossing Time:

Next to Last Threshold Crossing Time =

Transmit pulse waveform peak time +

The travel time between satellite and surface of the ground +

NextToLastThresholdLocation (within 150 sla02 waveform gates)-

LastPeakLocation (last peak location of sla02 waveform with in 150 gates)

offset 92 4ns Filter Peak Value: SLA02 last peak value.

offset 96 Peak Value for the selected filter: SLA02 last peak value.

offset 100 Time of the Peak Value for the selected filter:

Time of the Peak Value for the selected filter =

Transmit pulse waveform peak time +

The travel time between satellite and surface of the ground

offset 104 Filter Selected: set to 0.

offset 108 Threshold value, offset 112 Background Noise Mean Value for 4 ns filter, and

offset116 Background Noise Standard Deviation Value for the 4 ns filter: Threshold value = mean+4.5*sigma.

Mean and sigma are read in from sla02 data files.

offset 120 Range Window Status Word: set to 0.

offset 124 Calculated Weights for all Filters: set to 2000, 1000, 1000, 1000, 1000, 1000.

These values are not used in generate test data.

offset 148 Altimeter digitizer gain setting: set to 10.

offset 152, 154 Surface Echo Sample Padding and Compress Type: both set to 0.

offset 156 Surface Echo Data Samples: 544 samples for land, land/ice and 200 samples

for ocean, ocean/ice. IN REVERSED TIME ORDER!

**APID 15 & 17 Photon Counter & Cloud Digitizer Science Packet**

(from “*GLAS Atmospheric Channel Test Data Documentation*”; Steve Palm, Bill Hart, Judd Welton)

Simulated level 0 test data have been generated by the GLAS atmospheric lidar group for the express purpose of testing the I-SIPS atmospheric processing codes. This document first describes the process of the atmospheric data acquisition onboard the GLAS instrument and then discusses the content of the test data set and the methods used to generate it.

### GLAS Instrument Data Acquisition

The data acquired by GLAS range in height from 41 to –1 km for the 532 channel and 20 to –1 km for the 1064 channel. This height is with respect to the height above the local topography at the sub-satellite point. It is the job of the flight software to acquire the lidar data in such a way that the last bin of a profile is roughly 1 km below the surface. This is accomplished using an onboard 1x1 degree DEM (Digital Elevation Model) and an onboard calculation of the spacecraft altitude based on GPS (Global Positioning System) measurements. The equations which are evaluated onboard the spacecraft each second to calculate the 532 nm channel (PC) and the 1064 channel (CD) range gates at which to start taking data are:

(1 )

$$PC = H_{sat} - [H_{\min} - H_{off\ \min}] + PC_{bias}$$

(2)

$$CD = H_{sat} - [H_{\min} - H_{off\ \min}] + CD_{bias}$$

where Hsat is the height of the spacecraft, Hmin is the DEM minimum, Hoffmin is the offset associated with Hmin and PCbias and CDbias are the offsets to apply to the 532 and 1064 channels, respectively. Note that the only difference between equations 1 and 2 is the bias term, which can be different for each channel. Also note that even though the cloud digitizer board begins taking data at the same height (normally 41 km above the local DEM value) as the photon counting channel (assuming PCbias = CDbias), the flight software will only send down in the telemetry those 1064 nm data beginning 268 bins from this point (20.58 km). Hoffmin is set to a default of 1.0 km. The PC and CD biases can be used to move the profile either up (when made less than –41 km) or down (when made greater than –41 km). These will only be changed (from –41 km) for off-nadir pointing. The PC and CD values effectively represent the distance (in km) from the spacecraft to the top (first bin) of the data. They are known as the photon counter range gate and the cloud digitizer range gate. These equations are evaluated in real time (every second) aboard the spacecraft and the results are sent down in the telemetry data. The values reported in the telemetry are in units of time, not distance. Specifically, it is recorded as nanoseconds divided by 128, as shown in equations 3 and 4 below.

(3)

$$T_{pc} = PC/(c/2.0)10^9/128$$

(4)

$$T_{cd} = CD/(c/2.0)10^9/128$$

where c is the speed of light in km / s, and T represents time.

What this effectively means is that the lidar profiles can potentially shift up or down (with respect to the ellipsoid) from second to second in response to changes in the DEM. In the generation of our test data, we have mimicked this process of profile shifting by evaluating equations 1 through 4 at each second along the orbit. We used a preliminary version of the 1x1 degree DEM that will be used onboard the spacecraft. This step was actually essential to the generation of the test data set since one of the things that we want to test in the I-SIPS software is the vertical alignment (or de-shifting) of the lidar profiles (however, this step doesn't occur until level 1b processing).

Donghui Yi of code 971, supplied us with a preliminary GLAS orbit file which contained latitude, longitude, time and spacecraft height (Hsat) for 8 complete orbits. These data were given to us at a resolution of once every 10 seconds, and we linearly interpolated to once per second. Using the latitude and longitude we index into the DEM to obtain Hmin. The value of Hoffmin was 1.0 km, which means that when the actual DEM value was zero, we added 1.0 km to it and then divided by 0.125 to obtain the value stored at offset 1192 (see table 1 below). The value for both PCbias and CDbias was 41 km. Thus, at every second we calculated equation 1 through 4 to provide the correct values for the telemetry APPID 19, offset 586 (PC) and offset 598 (CD). In addition, the values of various parameters in the GPS / DEM section of APID 19 are also generated and given to Donghui via an ancillary file. These are shown in table 1 below.

**Table 3-5 The various parameters of telemetry APID 19 that are computed during the generation of the atmospheric test data**

| Offset | Parameter | Units |
|---|---|---|
| 586 | PC Range Gate | 128.0x10-9 seconds |
| 598 | CD Range Gate | 128.0x10-9 seconds |
| 1088 | Latitude | Degrees |
| 1090 | Longitude | Degrees |
| 1092 | Satellite Height | Km |
| 1116 | DEM minimum offset | Km |
| 1132 | PC Range bias | Km |
| 1136 | CD Range bias | Km |
| 1192 | DEM minimum | Meters/125 |

**Description of the Atmospheric Test Data Set**

The atmospheric test data set (APID's 15 and 17) is comprised of actual atmospheric observations made with a ground-based Micro Pulse Lidar (MPL). As such, they contain real cloud and aerosol structure. The original data file has been analyzed by Judd Welton of code 912, obtaining the top and bottom heights of the cloud and aerosol layers, the backscatter and extinction profiles, and the optical depth of the layers. Thus, when these test data are analyzed by the I-SIPS algorithms, the resulting output can be compared with the known answers. The original MPL data are first calibrated and the backscatter cross sections computed. These atmospheric cross sections are used as input to a GLAS simulation model which computes the signal that GLAS will measure.

We were asked to generate 8 orbits of test data that followed a predicted GLAS orbit supplied to us by the I-SIPS team. Because the original test data file (MPL) is about 6 minutes in length, the data had to be repeated many times over in order to cover the necessary 12 hour time span (8 orbits). This fact in no way presents a problem or diminishes its usefulness as a test data set. An example of a 30 minute segment of the data is shown in figure 1. This portion of data is from a segment of the orbit coming from the Pacific ocean and ending over Antarctica. One can see that the same data segment is repeated 3 times in this image. Also note that the coast of Antarctica is at about second 1500. At this point the DEM values become non zero and rapidly climb as we fly over the icesheet. As a result, the profiles are captured beginning at a higher altitude. However, when displayed in this fashion (height above the terrain), the profiles appear to shift downward in the image. Note the cloud feature, the top of which is at about 15 km when over the ocean (about second 1400). The same feature has dropped down to about 13 km when over Antarctica (about second 1700). This is exactly what will happen onboard the spacecraft.

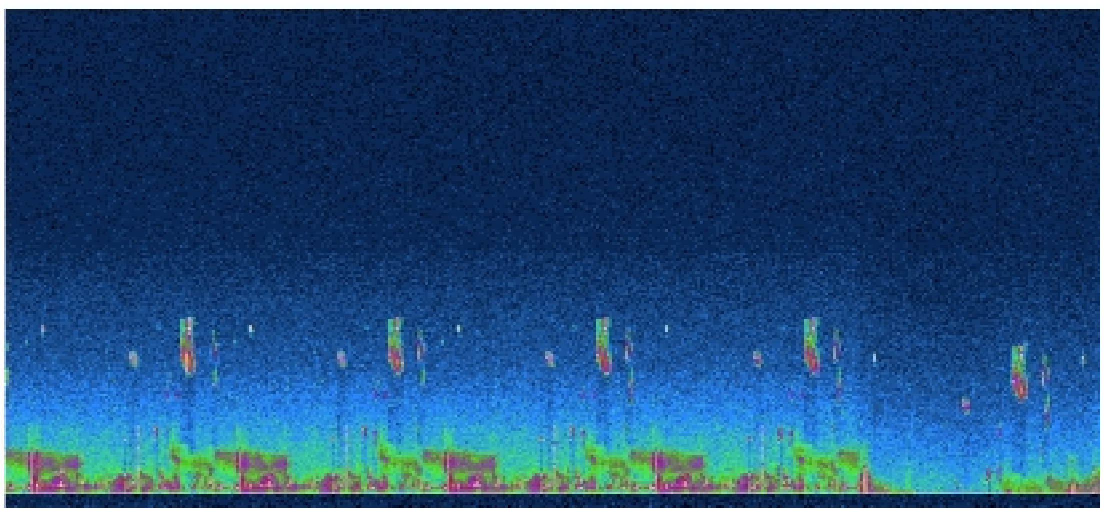

Figure 1. An image showing about 30 minutes of lidar test data along a segment of a GLAS predicted orbit starting in the southern Pacific Ocean and crossing over Antarctica. The vertical goes from 0 to 41 km.

### Other Required Values

In addition to the parameters described above, we need to include values for a number of other parameters in order to make the test data complete. These include laser energy, background values, the CD attenuation setting and dual pin A output. These values have been included in the test data set.

### Status

As of February 14, 2001, we have generated the first version (model 1a) of the level 0 atmospheric test data for both 1064 and 532 (APID's 15 and 17) and the associated APID 19 values listed in Table 1. There are some known problems with the 532 channel data which we would like to correct for future versions. In particular, the 532 data suffers from a round off problem that causes step jumps in the data at the boundaries of the various resolutions (at 20 km and 10 km). Also, the data above about 35 km are all zeroes, which is not realistic. We believe that the 1064 simulated data are good.

The data produced for model 1a is a nighttime simulation (background = 0), with no random noise, at the beginning of the mission. Also, there is no random noise added to the output signals. The 1064 data does, however, contain the correct detector noise associated with the APD detector.

Model 1b, produced on March 15, 2001 is a significant improvement on the model 1a output. While the input data (from the MPL) are the same (i.e., same atmospheric structure), we have added realistic shot noise generated using Poisson statistics to the data. A non-zero background was used (at a level consistent with day time background over the ocean), but it is everywhere constant. The 1064 data also have shot noise generated using Poisson statistics.

**Table 3-6 List of level 0 test data files delivered to I-SIPS for code testing to date**

| Date | Version | Data Source | Random Noise | Back-ground | Comments |
|---|---|---|---|---|---|
| 2/14/01 | Model 1a | MPL | No | Night | 8 orbits, 532 and 1064. Roundoff problem in 532. Limited clouds. No Stratospheric aerosols. Constant laser energy. |
| 2/15/01 | Model 1b | MPL | Yes | Day/Ocean | 8 orbits, 532 and 1064. Roundoff problem in 532 is fixed. Shot noise added. Constant background. Limited clouds. No Stratospheric aerosols. Constant laser energy. |

**Plan for Future Versions**

An improvement on model 1b would be to make the background level a function of the total column optical depth. This would make higher background over clouds, which is much more realistic than the current constant background. Test data where the laser energy fluctuates may also be useful. Further MPL measurements may also be added in the near future. Additionally, we have plans to use data acquired by the Cloud Physics Lidar (CPL) to generate longer test data sets. These will probably not be ready until early summer. Each version of data that is generated will be documented by updating section 4 of this document.

### 3.4.2 Detailed Description of GLA00_02 Input Data

GLA00_2 mostly matches the first 1000 seconds of GLA00_1. It is designed to test missing data within the APIDs. The science team has identified the following changes between the GLA00_2 and the first 1000 seconds of GLA00_1:

- For APID 19, the first 3 seconds and seconds 115, 116, 117, 118, and 119 data are missing.
- For APID 12/13, 6 packets are missing. They are packets 1000, 1001, 1002, 1003, 1004 and 1005.
- For APID 17, 3 seconds are missing. They are seconds 600, 601 and 602.
- For APID 15, 3 seconds are missing. They are seconds 800, 801 and 802.

### 3.4.3 Detailed Description of GLA00_03 Input Data

GLA00_3 mostly matches the time span from 2000-2100 seconds of GLA00_1. It is designed to test APID alignment splits at the PDS boundaries. The science team has identified the following changes between the GLA00_3 and the identified 1000 seconds of GLA00_1:

- For APID 19 and 15, the six hour boundary is the same as in the reference data.
- For APID 12/13, the six hour boundary is early by 0.5 second.
- For APID 17, the six hour boundary is early by 1 second.

### 3.4.4 Detailed Description of GLA00_04 Input Data

GLA00_4 mostly matches the time span from 2000-2100 seconds of GLA00_1. It is designed to test APID alignment splits at the PDS boundaries. The science team has identified the following changes between the GLA00_4 and the identified 1000 seconds of GLA00_1:

- For APID 17, the six hour boundary is the same as in the reference data.
- For APID 12/13, the six hour boundary is early by 0.5 second.
- For APID 19 and 15, the six hour boundary is late by 1 second.

Section 4

# Procedures

## 4.1 Environment

The GSAS software has been written and tested under the HP / UX 11.0 / Fortran 90 v2.3 environments. It is recommended that the acceptance test be run in an environment identical one of those used for development and testing. The team has identified problems with other versions of the HP compiler, so the version of the compiler is critical.

The team recommends, at a minimum, an HP L-class server with 1GB of RAM and 10 GB of free disk space.

## 4.2 Software Installation

Software should be installed and compiled as described in the GSAS User's Guide. This document will use the same conventions as the User's Guide with respect to directory designations. $GLAS_HOME will be designated as the directory in which the software was installed.

Integration testing was performed with no make options (neither DEBUG nor FAST were used). Acceptance testing should be performed with the same compiler flags.

## 4.3 Test Directory Setup

After the software has been installed, change to the $GLAS_HOME / test directory.

```
cd $GLAS_HOME/test
```

Run the setup script.

```
./setup_tests.sh
```

This script will create and populate six test directories, one for each of the six test cases. The contents of the test directories are identified in Section 3.4.

## 4.4 Recovery Steps

If a test procedure fails unexpectedly, the tester should look in the ANC06 file for error messages. Output products may also contain information as to the problem. Also verify the operating system and compiler versions.

Before re-running a test, the tester must remove or rename all output files created during the prior run. Scripts named “cleanup.sh’ have been provided in each test directory to do this.

If input files are accidently deleted, re-run the setup_tests.sh script to return to the default state.

If files were unrecoverably deleted, reinstall the software from the distribution media.

## 4.5 Test Cases

This section describes the six test cases of the GSAS Acceptance Test. See Appendix A of the GSAS User's Guide to identify input and output file types. Additional debug input/outputs may be created in each test if differences are found in the output binary files.

An important note is that tests which require output files normally created by a previous test actually use the **reference** files for that test. This preserves the independence of each test, but requires reference files to be replaced if data needs to be updated.

The run_all.sh script will run each test in sequence. It is recommended that stdout be redirected to a file in order to capture the results of each test. Instructions are provided in the next sections for running each test individually, if required.

### 4.5.1 Test 1: GLAS_L0Proc Processing

This test uses 25 hours of known GLA00 data to create ANC29 and ANC32 files correlating to each input dataset. All input files will have been linked into the test directory by the setup shell script. This test case takes about an hour to run on an HP L-class server.

#### 4.5.1.1 Inputs

```
anc07_001_01_00.dat (error constants)
anc07_001_01_01.dat (global constants)
anc25_002_20000101_000000_01_00.dat (GPS/UTC time conversion table)
anc33_002_20000101_000000_01_00.dat (MET/UTC time conversion table)
(unique control files for each test)
(GLA00_01)
gla00_002_20000101_000000_01_12.dat
gla00_002_20000101_000000_01_13.dat
gla00_002_20000101_000000_01_15.dat
gla00_002_20000101_000000_01_17.dat
gla00_002_20000101_000000_01_19.dat
gla00_002_20000101_010000_01_12.dat
gla00_002_20000101_010000_01_13.dat
gla00_002_20000101_010000_01_15.dat
gla00_002_20000101_010000_01_17.dat
gla00_002_20000101_010000_01_19.dat
gla00_002_20000101_020000_01_12.dat
gla00_002_20000101_020000_01_13.dat
gla00_002_20000101_020000_01_15.dat
gla00_002_20000101_020000_01_17.dat
gla00_002_20000101_020000_01_19.dat
gla00_002_20000101_030000_01_12.dat
gla00_002_20000101_030000_01_13.dat
gla00_002_20000101_030000_01_15.dat
gla00_002_20000101_030000_01_17.dat
gla00_002_20000101_030000_01_19.dat
gla00_002_20000101_040000_01_12.dat
```

```
gla00_002_20000101_040000_01_13.dat
gla00_002_20000101_040000_01_15.dat
gla00_002_20000101_040000_01_17.dat
gla00_002_20000101_040000_01_19.dat
(GLA00_02)
gla00_002_20000101_000000_02_12.dat
gla00_002_20000101_000000_02_13.dat
gla00_002_20000101_000000_02_15.dat
gla00_002_20000101_000000_02_17.dat
gla00_002_20000101_000000_02_19.dat
(GLA00_03)
gla00_002_20000101_000000_03_12.dat
gla00_002_20000101_000000_03_13.dat
gla00_002_20000101_000000_03_15.dat
gla00_002_20000101_000000_03_17.dat
gla00_002_20000101_000000_03_19.dat
gla00_002_20000101_010000_03_12.dat
gla00_002_20000101_010000_03_13.dat
gla00_002_20000101_010000_03_15.dat
gla00_002_20000101_010000_03_17.dat
gla00_002_20000101_010000_03_19.dat
(GLA00_04)
gla00_002_20000101_000000_04_12.dat
gla00_002_20000101_000000_04_13.dat
gla00_002_20000101_000000_04_15.dat
gla00_002_20000101_000000_04_17.dat
gla00_002_20000101_000000_04_19.dat
gla00_002_20000101_010000_04_12.dat
gla00_002_20000101_010000_04_13.dat
gla00_002_20000101_010000_04_15.dat
gla00_002_20000101_010000_04_17.dat
gla00_002_20000101_010000_04_19.dat
```

#### 4.5.1.2 Outputs

The sets of ANC29 and ANC32 output files created during this procedure will correspond to the set of GLA00_x input files used to create them and be named accordingly. Each anc06 and stdout file created will be named according to the phase and test number. For example, the anc06 file name for Phase 2, Test 3 is anc06_p2t3.dat. Multiple output files within the same test are identified with subsequent letters. For example, p2t3a.stdout, p2t3b.stdout, p2t3c.stdout.

```
(unique anc06 and stdout files for each test)
The GLA00_1 output dataset contains the following ANC29/32 files:
anc29_002_20000101_000000_01_00.dat
anc29_002_20000101_010000_01_00.dat
anc29_002_20000101_020000_01_00.dat
anc29_002_20000101_030000_01_00.dat
anc29_002_20000101_040000_01_00.dat
anc32_002_20000101_000000_01_00.dat
anc32_002_20000101_010000_01_00.dat
anc32_002_20000101_020000_01_00.dat
anc32_002_20000101_030000_01_00.dat
anc32_002_20000101_040000_01_00.dat
The GLA00_2 output dataset contains the following ANC29/32 files:
anc29_002_20000101_000000_02_00.dat
anc32_002_20000101_000000_02_00.dat
The GLA00_3 output dataset contains the following ANC29/32 files:
```

```
anc29_002_20000101_000000_03_00.dat
anc29_002_20000101_010000_03_00.dat
anc32_002_20000101_000000_03_00.dat
anc32_002_20000101_010000_03_00.dat
The GLA00_4 output dataset contains the following ANC29/32 files:
anc29_002_20000101_000000_04_00.dat
anc29_002_20000101_010000_04_00.dat
anc32_002_20000101_000000_04_00.dat
anc32_002_20000101_010000_04_00.dat
```

#### 4.5.1.3 Procedure

In order to run this test individually, change to the $GLAS_HOME/test/test1 directory and run the test script.

```
cd $GLAS_HOME/test/test1
./run.sh
```

Examine the output to verify the results are successful and that the newly-created test data match the reference data.

### 4.5.2 Test 2: L1A Processing

This test uses 25 hours of known GLA00 data and the **reference** ANC29/32 files to create multi-granule GLA01-02. It tests the functionality of using ANC29 files to synchronously read GLA00 APIDs and writing multiple granules of GLA01-02. All input files will have been linked into the test directory by the setup shell script. This test case takes about 3 hours to run on an HP L-class server.

#### 4.5.2.1 Inputs

```
anc07_001_01_00.dat (error constants)
anc07_001_01_01.dat (global constants)
anc07_001_01_05.dat (l1a constants)
anc25_002_20000101_000000_01_00.dat (GPS/UTC time conversion table)
anc33_002_20000101_000000_01_00.dat (MET/UTC time conversion table)
anc20_002_20000101_000000_01_00.dat (predicted orbit file)
(unique control files for each test)
(GLA00_01 as described in test 1)
(GLA00_02 as described in test 1)
(GLA00_03 as described in test 1)
(GLA00_04 as described in test 1)
```

#### 4.5.2.2 Outputs

The sets of ANC29 and ANC32 output files created during this procedure will correspond to the set of GLA00_x input files used to create them and be named accordingly. Each anc06 and stdout file created will be named according to the phase and test number. For example, the anc06 file name for Phase 2, Test 3 is anc06_p2t3.dat. Multiple output files within the same test are identified with subsequent letters. For example, p2t3a.stdout, p2t3b.stdout, p2t3c.stdout.

```
(unique anc06 and stdout files for each test)
The GLA01_01 output dataset contains the following files:
gla01_002_11_0001_0028_2_00_00.dat
gla01_002_11_0001_0028_3_00_00.dat
gla01_002_11_0001_0028_4_00_00.dat
gla01_002_11_0001_0029_1_00_00.dat
```

```
gla01_002_11_0001_0029_2_00_00.dat
gla01_002_11_0001_0029_3_00_00.dat
gla01_002_11_0001_0029_4_00_00.dat
gla01_002_11_0001_0030_1_00_00.dat
gla01_002_11_0001_0030_2_00_00.dat
gla01_002_11_0001_0030_3_00_00.dat
gla01_002_11_0001_0030_4_00_00.dat
gla01_002_11_0001_0031_1_00_00.dat
gla01_002_11_0001_0031_2_00_00.dat
gla01_002_11_0001_0031_3_00_00.dat
gla01_002_11_0001_0031_4_00_00.dat
gla01_002_11_0001_0032_1_00_00.dat
gla01_002_11_0001_0032_2_00_00.dat
gla01_002_11_0001_0032_3_00_00.dat
gla01_002_11_0001_0032_4_00_00.dat
gla01_002_11_0001_0033_1_00_00.dat
gla01_002_11_0001_0033_2_00_00.dat
gla01_002_11_0001_0033_3_00_00.dat
gla01_002_11_0001_0033_4_00_00.dat
gla01_002_11_0001_0034_1_00_00.dat
gla01_002_11_0001_0034_2_00_00.dat
gla01_002_11_0001_0034_3_00_00.dat
gla01_002_11_0001_0034_4_00_00.dat
gla01_002_11_0001_0035_1_00_00.dat
gla01_002_11_0001_0035_2_00_00.dat
gla01_002_11_0001_0035_3_00_00.dat
gla01_002_11_0001_0035_4_00_00.dat
gla01_002_11_0001_0036_1_00_00.dat
gla01_002_11_0001_0036_2_00_00.dat
gla01_002_11_0001_0036_3_00_00.dat
gla01_002_11_0001_0036_4_00_00.dat
gla01_002_11_0001_0037_1_00_00.dat
gla01_002_11_0001_0037_2_00_00.dat
gla01_002_11_0001_0037_3_00_00.dat
gla01_002_11_0001_0037_4_00_00.dat
gla01_002_11_0001_0038_1_00_00.dat
gla01_002_11_0001_0038_2_00_00.dat
gla01_002_11_0001_0038_3_00_00.dat
gla01_002_11_0001_0038_4_00_00.dat
gla01_002_11_0001_0039_1_00_00.dat
gla01_002_11_0001_0039_2_00_00.dat
gla01_002_11_0001_0039_3_00_00.dat
gla01_002_11_0001_0039_4_00_00.dat
gla01_002_11_0001_0040_1_00_00.dat
gla01_002_11_0001_0040_2_00_00.dat
gla01_002_11_0001_0040_3_00_00.dat
gla01_002_11_0001_0040_4_00_00.dat
gla01_002_11_0001_0041_1_00_00.dat
gla01_002_11_0001_0041_2_00_00.dat
gla01_002_11_0001_0041_3_00_00.dat
gla01_002_11_0001_0041_4_00_00.dat
gla01_002_11_0001_0042_1_00_00.dat
gla01_002_11_0001_0042_2_00_00.dat
gla01_002_11_0001_0042_3_00_00.dat
gla01_002_11_0001_0042_4_00_00.dat
gla01_002_11_0001_0043_1_00_00.dat
gla01_002_11_0001_0043_2_00_00.dat
gla01_002_11_0001_0043_3_00_00.dat
gla01_002_11_0001_0043_4_00_00.dat
```

```
qap01_002_11_0001_0028_2_00_00.dat
qap01_002_11_0001_0028_3_00_00.dat
qap01_002_11_0001_0028_4_00_00.dat
qap01_002_11_0001_0029_1_00_00.dat
qap01_002_11_0001_0029_2_00_00.dat
qap01_002_11_0001_0029_3_00_00.dat
qap01_002_11_0001_0029_4_00_00.dat
qap01_002_11_0001_0030_1_00_00.dat
qap01_002_11_0001_0030_2_00_00.dat
qap01_002_11_0001_0030_3_00_00.dat
qap01_002_11_0001_0030_4_00_00.dat
qap01_002_11_0001_0031_1_00_00.dat
qap01_002_11_0001_0031_2_00_00.dat
qap01_002_11_0001_0031_3_00_00.dat
qap01_002_11_0001_0031_4_00_00.dat
qap01_002_11_0001_0032_1_00_00.dat
qap01_002_11_0001_0032_2_00_00.dat
qap01_002_11_0001_0032_3_00_00.dat
qap01_002_11_0001_0032_4_00_00.dat
qap01_002_11_0001_0033_1_00_00.dat
qap01_002_11_0001_0033_2_00_00.dat
qap01_002_11_0001_0033_3_00_00.dat
qap01_002_11_0001_0033_4_00_00.dat
qap01_002_11_0001_0034_1_00_00.dat
qap01_002_11_0001_0034_2_00_00.dat
qap01_002_11_0001_0034_3_00_00.dat
qap01_002_11_0001_0034_4_00_00.dat
qap01_002_11_0001_0035_1_00_00.dat
qap01_002_11_0001_0035_2_00_00.dat
qap01_002_11_0001_0035_3_00_00.dat
qap01_002_11_0001_0035_4_00_00.dat
qap01_002_11_0001_0036_1_00_00.dat
qap01_002_11_0001_0036_2_00_00.dat
qap01_002_11_0001_0036_3_00_00.dat
qap01_002_11_0001_0036_4_00_00.dat
qap01_002_11_0001_0037_1_00_00.dat
qap01_002_11_0001_0037_2_00_00.dat
qap01_002_11_0001_0037_3_00_00.dat
qap01_002_11_0001_0037_4_00_00.dat
qap01_002_11_0001_0038_1_00_00.dat
qap01_002_11_0001_0038_2_00_00.dat
qap01_002_11_0001_0038_3_00_00.dat
qap01_002_11_0001_0038_4_00_00.dat
qap01_002_11_0001_0039_1_00_00.dat
qap01_002_11_0001_0039_2_00_00.dat
qap01_002_11_0001_0039_3_00_00.dat
qap01_002_11_0001_0039_4_00_00.dat
qap01_002_11_0001_0040_1_00_00.dat
qap01_002_11_0001_0040_2_00_00.dat
qap01_002_11_0001_0040_3_00_00.dat
qap01_002_11_0001_0040_4_00_00.dat
qap01_002_11_0001_0041_1_00_00.dat
qap01_002_11_0001_0041_2_00_00.dat
qap01_002_11_0001_0041_3_00_00.dat
qap01_002_11_0001_0041_4_00_00.dat
qap01_002_11_0001_0042_1_00_00.dat
qap01_002_11_0001_0042_2_00_00.dat
qap01_002_11_0001_0042_3_00_00.dat
qap01_002_11_0001_0042_4_00_00.dat
```

```
qap01_002_11_0001_0043_1_00_00.dat
qap01_002_11_0001_0043_2_00_00.dat
qap01_002_11_0001_0043_3_00_00.dat
qap01_002_11_0001_0043_4_00_00.dat
The GLA02_01 output dataset contains the following files:
gla02_002_11_0001_0027_0_00_00.dat
gla02_002_11_0001_0029_0_00_00.dat
gla02_002_11_0001_0031_0_00_00.dat
gla02_002_11_0001_0033_0_00_00.dat
gla02_002_11_0001_0035_0_00_00.dat
gla02_002_11_0001_0037_0_00_00.dat
gla02_002_11_0001_0039_0_00_00.dat
gla02_002_11_0001_0041_0_00_00.dat
gla02_002_11_0001_0043_0_00_00.dat
qap02_002_11_0001_0027_0_00_00.dat
qap02_002_11_0001_0029_0_00_00.dat
qap02_002_11_0001_0031_0_00_00.dat
qap02_002_11_0001_0033_0_00_00.dat
qap02_002_11_0001_0035_0_00_00.dat
qap02_002_11_0001_0037_0_00_00.dat
qap02_002_11_0001_0039_0_00_00.dat
qap02_002_11_0001_0041_0_00_00.dat
qap02_002_11_0001_0043_0_00_00.dat
The GLA01_02 output dataset contains the following files:
gla01_002_11_0001_0028_2_02_00.dat
gla01_002_11_0001_0028_3_02_00.dat
qap01_002_11_0001_0028_2_02_00.dat
qap01_002_11_0001_0028_3_02_00.dat
The GLA02_02 output dataset contains the following files:
gla02_002_11_0001_0027_0_02_00.dat
qap02_002_11_0001_0027_0_02_00.dat
The GLA01_03 output dataset contains the following files:
gla01_002_11_0001_0031_4_03_00.dat
qap01_002_11_0001_0031_4_03_00.dat
The GLA02_03 output dataset contains the following files:
gla02_002_11_0001_0031_0_03_00.dat
qap02_002_11_0001_0031_0_03_00.dat
The GLA01_04 output dataset contains the following files:
gla01_002_11_0001_0031_4_04_00.dat
qap01_002_11_0001_0031_4_04_00.dat
The GLA02_04 output dataset contains the following files:
gla02_002_11_0001_0031_0_04_00.dat
qap02_002_11_0001_0031_0_04_00.dat
```

#### 4.5.2.3 Procedure

In order to run this test individually, change to the $GLAS_HOME/test/test2 directory and run the test script.

```
cd $GLAS_HOME/test/test2
./run.sh
```

Examine the output to verify the results are successful and that the newly-created test data match the reference data.

### 4.5.3 Test 3: GSAS Utilities

This test exercises the following GSAS utilities: createGran_util, reforbit_util, met_util and atm_anc. Output files created by met_util and atm_anc are needed for downstream processing.

All input files will have been linked into the test directory by the setup shell script. This test case takes approximately 0.5 hours to run on an HP L-class server.

#### 4.5.3.1 Inputs

```
(unique control files for each test)
(GLA02_01)
anc26_000_20000101_120000_01.dat
anc07_001_01_00.dat
anc07_001_01_06.dat
reforbID_p3t1.dat
v2_8dayorb.da
anc01_001_20000101_000000_01.dat
anc01_001_20000101_060000_01.dat
anc01_001_20000101_120000_01.dat
anc01_001_20000101_180000_01.dat
anc01_001_20000102_000000_01.dat
anc01_001_20000102_060000_01.dat
anc01_001_20000102_120000_01.dat
anc01_001_20000102_180000_01.dat
```

#### 4.5.3.2 Outputs

```
(unique stdout files for each test)
anc22_002_20000101_120000_01.dat
gran_p3t2_qrt_revfile.dat
gran_p3t2_two_revfile.dat
gran_p3t2_fteen_revfile.dat
gran_p3t2_scf_revfile.dat
The ANC01_01 output dataset contains the following files:
anc01_001_20000101_000000_01_00.dat
anc01_001_20000101_000000_01_01.dat
anc01_001_20000101_000000_01_02.dat
anc01_001_20000101_000000_01_03.dat
anc01_001_20000101_000000_01_04.dat
anc01_001_20000101_060000_01_00.dat
anc01_001_20000101_060000_01_01.dat
anc01_001_20000101_060000_01_02.dat
anc01_001_20000101_060000_01_03.dat
anc01_001_20000101_060000_01_04.dat
anc01_001_20000101_120000_01_00.dat
anc01_001_20000101_120000_01_01.dat
anc01_001_20000101_120000_01_02.dat
anc01_001_20000101_120000_01_03.dat
anc01_001_20000101_120000_01_04.dat
anc01_001_20000101_180000_01_00.dat
anc01_001_20000101_180000_01_01.dat
anc01_001_20000101_180000_01_02.dat
anc01_001_20000101_180000_01_03.dat
anc01_001_20000101_180000_01_04.dat
anc01_001_20000102_000000_01_00.dat
anc01_001_20000102_000000_01_01.dat
```

```
anc01_001_20000102_000000_01_02.dat
anc01_001_20000102_000000_01_03.dat
anc01_001_20000102_000000_01_04.dat
anc01_001_20000102_060000_01_00.dat
anc01_001_20000102_060000_01_01.dat
anc01_001_20000102_060000_01_02.dat
anc01_001_20000102_060000_01_03.dat
anc01_001_20000102_060000_01_04.dat
anc01_001_20000102_120000_01_00.dat
anc01_001_20000102_120000_01_01.dat
anc01_001_20000102_120000_01_02.dat
anc01_001_20000102_120000_01_03.dat
anc01_001_20000102_120000_01_04.dat
anc01_001_20000102_180000_01_00.dat
anc01_001_20000102_180000_01_01.dat
anc01_001_20000102_180000_01_02.dat
anc01_001_20000102_180000_01_03.dat
anc01_001_20000102_180000_01_04.dat
The ANC36_01 output dataset contains the following files:
anc36_002_11_0001_0027_0_00_00.dat
anc36_002_11_0001_0029_0_00_00.dat
anc36_002_11_0001_0031_0_00_00.dat
anc36_002_11_0001_0033_0_00_00.dat
anc36_002_11_0001_0035_0_00_00.dat
anc36_002_11_0001_0037_0_00_00.dat
anc36_002_11_0001_0039_0_00_00.dat
anc36_002_11_0001_0041_0_00_00.dat
anc36_002_11_0001_0043_0_00_00.dat
```

#### 4.5.3.3 Procedure

In order to run this test individually, change to the $GLAS_HOME/test/test3 directory and run the test script.

```
cd $GLAS_HOME/test/test3
./run.sh
```

Examine the output to verify the results are successful and that the newly-created test data match the reference data.

### 4.5.4 Test 4: Atmosphere Processing

This test uses the GLA02_01, ANC36_01, and ANC01_01 **reference** data to create multi-granule GLA07-09_01. This test exercises the Atmosphere subsystem. GLA10 and GLA11 are not supported.

All input files will have been linked into the test directory by the setup shell script. This test case takes approximately 12 minutes to run on an HP L-class server.

#### 4.5.4.1 Inputs

```
(unique control files for each test)
(GLA02_01)
(ANC01_01)
(ANC36_01)
anc07_001_01_00.dat
anc07_001_01_01.dat
anc07_001_01_02.dat
```

```
anc07_001_01_05.dat
anc08_001_20000101_000000_01_00.dat
anc09_001_20000101_000000_01_00.dat
anc12_001_01_00.dat
anc12_001_01_01.dat
anc13_001_01_00.dat
anc18_001_01_00.dat
anc24_001_20000101_120000_01_00.dat
anc30_001_01_00.dat
anc31_001_01_00.dat
anc35_001_01_00.dat
```

### 4.5.4.2 Outputs

```
(unique anc06 and stdout files for each test)
The GLA07-09_01 output dataset contains the following
files:gla07_002_11_0001_0027_0_00_00.dat
gla07_002_11_0001_0029_0_00_00.dat
gla07_002_11_0001_0031_0_00_00.dat
gla07_002_11_0001_0033_0_00_00.dat
gla07_002_11_0001_0035_0_00_00.dat
gla07_002_11_0001_0037_0_00_00.dat
gla07_002_11_0001_0039_0_00_00.dat
gla07_002_11_0001_0041_0_00_00.dat
gla07_002_11_0001_0043_0_00_00.dat
gla08_002_11_0001_0015_0_00_00.dat
gla08_002_11_0001_0029_0_00_00.dat
gla08_002_11_0001_0043_0_00_00.dat
gla09_002_11_0001_0015_0_00_00.dat
gla09_002_11_0001_0029_0_00_00.dat
gla09_002_11_0001_0043_0_00_00.dat
qap07_002_11_0001_0027_0_00_00.dat
qap07_002_11_0001_0029_0_00_00.dat
qap07_002_11_0001_0031_0_00_00.dat
qap07_002_11_0001_0033_0_00_00.dat
qap07_002_11_0001_0035_0_00_00.dat
qap07_002_11_0001_0037_0_00_00.dat
qap07_002_11_0001_0039_0_00_00.dat
qap07_002_11_0001_0041_0_00_00.dat
qap07_002_11_0001_0043_0_00_00.dat
qap08_002_11_0001_0015_0_00_00.dat
qap08_002_11_0001_0029_0_00_00.dat
qap08_002_11_0001_0043_0_00_00.dat
qap09_002_11_0001_0015_0_00_00.dat
qap09_002_11_0001_0029_0_00_00.dat
qap09_002_11_0001_0043_0_00_00.dat
```

### 4.5.4.3 Procedure

In order to run this test individually, change to the $GLAS_HOME/test/test4 directory and run the test script.

```
cd $GLAS_HOME/test/test4
./run.sh
```

Examine the output to verify the results are successful and that the newly-created test data match the reference data.

### 4.5.5 Test 5: Waveform Processing

This test uses the GLA01_01 **reference** data to create multi-granule GLA05_01. This test exercises the waveforms subsystem.

All input files will have been linked into the test directory by the setup shell script. This test case takes approximately 12 hours to run on an HP L-class server.

#### 4.5.5.1 Inputs

```
(unique control files for each test)
(GLA01_01)
anc07_001_01_00.dat
anc07_001_01_01.dat
anc07_001_01_04.dat
anc08_001_20000101_000000_01_00.dat
anc09_001_20000101_000000_01_00.dat
anc24_001_20000101_120000_01_00.dat
anc27_001_01_00.dat
anc27_001_01_01.dat
```

#### 4.5.5.2 Outputs

```
(unique anc06 and stdout files for each test)
The GLA05_01 output dataset contains the following files:
gla05_002_11_0001_0028_2_00_00.dat
gla05_002_11_0001_0028_3_00_00.dat
gla05_002_11_0001_0028_4_00_00.dat
gla05_002_11_0001_0029_1_00_00.dat
gla05_002_11_0001_0029_2_00_00.dat
gla05_002_11_0001_0029_3_00_00.dat
gla05_002_11_0001_0029_4_00_00.dat
gla05_002_11_0001_0030_1_00_00.dat
gla05_002_11_0001_0030_2_00_00.dat
gla05_002_11_0001_0030_3_00_00.dat
gla05_002_11_0001_0030_4_00_00.dat
gla05_002_11_0001_0031_1_00_00.dat
gla05_002_11_0001_0031_2_00_00.dat
gla05_002_11_0001_0031_3_00_00.dat
gla05_002_11_0001_0031_4_00_00.dat
gla05_002_11_0001_0032_1_00_00.dat
gla05_002_11_0001_0032_2_00_00.dat
gla05_002_11_0001_0032_3_00_00.dat
gla05_002_11_0001_0032_4_00_00.dat
gla05_002_11_0001_0033_1_00_00.dat
gla05_002_11_0001_0033_2_00_00.dat
gla05_002_11_0001_0033_3_00_00.dat
gla05_002_11_0001_0033_4_00_00.dat
gla05_002_11_0001_0034_1_00_00.dat
gla05_002_11_0001_0034_2_00_00.dat
gla05_002_11_0001_0034_3_00_00.dat
gla05_002_11_0001_0034_4_00_00.dat
gla05_002_11_0001_0035_1_00_00.dat
gla05_002_11_0001_0035_2_00_00.dat
gla05_002_11_0001_0035_3_00_00.dat
gla05_002_11_0001_0035_4_00_00.dat
gla05_002_11_0001_0036_1_00_00.dat
gla05_002_11_0001_0036_2_00_00.dat
```

```
gla05_002_11_0001_0036_3_00_00.dat
gla05_002_11_0001_0036_4_00_00.dat
gla05_002_11_0001_0037_1_00_00.dat
gla05_002_11_0001_0037_2_00_00.dat
gla05_002_11_0001_0037_3_00_00.dat
gla05_002_11_0001_0037_4_00_00.dat
gla05_002_11_0001_0038_1_00_00.dat
gla05_002_11_0001_0038_2_00_00.dat
gla05_002_11_0001_0038_3_00_00.dat
gla05_002_11_0001_0038_4_00_00.dat
gla05_002_11_0001_0039_1_00_00.dat
gla05_002_11_0001_0039_2_00_00.dat
gla05_002_11_0001_0039_3_00_00.dat
gla05_002_11_0001_0039_4_00_00.dat
gla05_002_11_0001_0040_1_00_00.dat
gla05_002_11_0001_0040_2_00_00.dat
gla05_002_11_0001_0040_3_00_00.dat
gla05_002_11_0001_0040_4_00_00.dat
gla05_002_11_0001_0041_1_00_00.dat
gla05_002_11_0001_0041_2_00_00.dat
gla05_002_11_0001_0041_3_00_00.dat
gla05_002_11_0001_0041_4_00_00.dat
gla05_002_11_0001_0042_1_00_00.dat
gla05_002_11_0001_0042_2_00_00.dat
gla05_002_11_0001_0042_3_00_00.dat
gla05_002_11_0001_0042_4_00_00.dat
gla05_002_11_0001_0043_1_00_00.dat
gla05_002_11_0001_0043_2_00_00.dat
gla05_002_11_0001_0043_3_00_00.dat
gla05_002_11_0001_0043_4_00_00.dat
qap05_002_11_0001_0028_2_00_00.dat
qap05_002_11_0001_0028_3_00_00.dat
qap05_002_11_0001_0028_4_00_00.dat
qap05_002_11_0001_0029_1_00_00.dat
qap05_002_11_0001_0029_2_00_00.dat
qap05_002_11_0001_0029_3_00_00.dat
qap05_002_11_0001_0029_4_00_00.dat
qap05_002_11_0001_0030_1_00_00.dat
qap05_002_11_0001_0030_2_00_00.dat
qap05_002_11_0001_0030_3_00_00.dat
qap05_002_11_0001_0030_4_00_00.dat
qap05_002_11_0001_0031_1_00_00.dat
qap05_002_11_0001_0031_2_00_00.dat
qap05_002_11_0001_0031_3_00_00.dat
qap05_002_11_0001_0031_4_00_00.dat
qap05_002_11_0001_0032_1_00_00.dat
qap05_002_11_0001_0032_2_00_00.dat
qap05_002_11_0001_0032_3_00_00.dat
qap05_002_11_0001_0032_4_00_00.dat
qap05_002_11_0001_0033_1_00_00.dat
qap05_002_11_0001_0033_2_00_00.dat
qap05_002_11_0001_0033_3_00_00.dat
qap05_002_11_0001_0033_4_00_00.dat
qap05_002_11_0001_0034_1_00_00.dat
qap05_002_11_0001_0034_2_00_00.dat
qap05_002_11_0001_0034_3_00_00.dat
qap05_002_11_0001_0034_4_00_00.dat
qap05_002_11_0001_0035_1_00_00.dat
qap05_002_11_0001_0035_2_00_00.dat
```

```
qap05_002_11_0001_0035_3_00_00.dat
qap05_002_11_0001_0035_4_00_00.dat
qap05_002_11_0001_0036_1_00_00.dat
qap05_002_11_0001_0036_2_00_00.dat
qap05_002_11_0001_0036_3_00_00.dat
qap05_002_11_0001_0036_4_00_00.dat
qap05_002_11_0001_0037_1_00_00.dat
qap05_002_11_0001_0037_2_00_00.dat
qap05_002_11_0001_0037_3_00_00.dat
qap05_002_11_0001_0037_4_00_00.dat
qap05_002_11_0001_0038_1_00_00.dat
qap05_002_11_0001_0038_2_00_00.dat
qap05_002_11_0001_0038_3_00_00.dat
qap05_002_11_0001_0038_4_00_00.dat
qap05_002_11_0001_0039_1_00_00.dat
qap05_002_11_0001_0039_2_00_00.dat
qap05_002_11_0001_0039_3_00_00.dat
qap05_002_11_0001_0039_4_00_00.dat
qap05_002_11_0001_0040_1_00_00.dat
qap05_002_11_0001_0040_2_00_00.dat
qap05_002_11_0001_0040_3_00_00.dat
qap05_002_11_0001_0040_4_00_00.dat
qap05_002_11_0001_0041_1_00_00.dat
qap05_002_11_0001_0041_2_00_00.dat
qap05_002_11_0001_0041_3_00_00.dat
qap05_002_11_0001_0041_4_00_00.dat
qap05_002_11_0001_0042_1_00_00.dat
qap05_002_11_0001_0042_2_00_00.dat
qap05_002_11_0001_0042_3_00_00.dat
qap05_002_11_0001_0042_4_00_00.dat
qap05_002_11_0001_0043_1_00_00.dat
qap05_002_11_0001_0043_2_00_00.dat
qap05_002_11_0001_0043_3_00_00.dat
qap05_002_11_0001_0043_4_00_00.dat
```

#### 4.5.5.3 Procedure

In order to run this test individually, change to the $GLAS_HOME/test/test5 directory and run the test script.

```
cd $GLAS_HOME/test/test5
./run.sh
```

Examine the output to verify the results are successful and that the newly-created test data match the reference data.

### 4.5.6 Test 6: Elevation Processing

This test uses the GLA05_01 **reference** data to create the output GLA06,12-15_01 output dataset. This test exercises the elevations subsystem.

All input files will have been linked into the test directory by the setup shell script. This test case takes approximately 1 hour and 20 minutes to run on an HP L-class server.

#### 4.5.6.1 Inputs

```
cf05_001_20001027_001.ctl
anc01_001_20000101_120000_01_01.dat
anc01_001_20000101_120000_01_02.dat
anc01_001_20000101_120000_01_03.dat
anc01_001_20000101_120000_01_04.dat
anc01_001_20000101_180000_01_00.dat
anc01_001_20000101_180000_01_01.dat
anc01_001_20000101_180000_01_02.dat
anc01_001_20000101_180000_01_03.dat
anc01_001_20000101_180000_01_04.dat
anc07_001_01_00.dat
anc07_001_01_01.dat
anc07_001_01_03.dat
anc08_001_20000101_000000_01_00.dat
anc09_001_20000101_000000_01_00.dat
anc12_001_01_00.dat
anc12_001_01_01.dat
anc13_001_01_00.dat
anc16_001_01_00.dat
anc17_001_01_00.dat
anc18_001_01_00.dat
anc24_001_20000101_120000_01_00.dat
anc30_001_01_00.dat
anc31_001_01_00.dat
gla05_001_11_001_0001_1_01_00.dat
gla05_001_11_001_0001_2_01_00.dat
gla05_001_11_001_0001_3_01_00.dat
```

#### 4.5.6.2 Outputs

```
(unique anc06 and stdout files for each test)
The GLA06,12-15_01 output dataset contains the following files:
gla06_002_11_0001_0028_2_00_00.dat
gla06_002_11_0001_0028_3_00_00.dat
gla06_002_11_0001_0028_4_00_00.dat
gla06_002_11_0001_0029_1_00_00.dat
gla06_002_11_0001_0029_2_00_00.dat
gla06_002_11_0001_0029_3_00_00.dat
gla06_002_11_0001_0029_4_00_00.dat
gla06_002_11_0001_0030_1_00_00.dat
gla06_002_11_0001_0030_2_00_00.dat
gla06_002_11_0001_0030_3_00_00.dat
gla06_002_11_0001_0030_4_00_00.dat
gla06_002_11_0001_0031_1_00_00.dat
gla06_002_11_0001_0031_2_00_00.dat
gla06_002_11_0001_0031_3_00_00.dat
gla06_002_11_0001_0031_4_00_00.dat
gla06_002_11_0001_0032_1_00_00.dat
gla06_002_11_0001_0032_2_00_00.dat
gla06_002_11_0001_0032_3_00_00.dat
gla06_002_11_0001_0032_4_00_00.dat
gla06_002_11_0001_0033_1_00_00.dat
gla06_002_11_0001_0033_2_00_00.dat
gla06_002_11_0001_0033_3_00_00.dat
gla06_002_11_0001_0033_4_00_00.dat
```

```
gla06_002_11_0001_0034_1_00_00.dat
gla06_002_11_0001_0034_2_00_00.dat
gla06_002_11_0001_0034_3_00_00.dat
gla06_002_11_0001_0034_4_00_00.dat
gla06_002_11_0001_0035_1_00_00.dat
gla06_002_11_0001_0035_2_00_00.dat
gla06_002_11_0001_0035_3_00_00.dat
gla06_002_11_0001_0035_4_00_00.dat
gla06_002_11_0001_0036_1_00_00.dat
gla06_002_11_0001_0036_2_00_00.dat
gla06_002_11_0001_0036_3_00_00.dat
gla06_002_11_0001_0036_4_00_00.dat
gla06_002_11_0001_0037_1_00_00.dat
gla06_002_11_0001_0037_2_00_00.dat
gla06_002_11_0001_0037_3_00_00.dat
gla06_002_11_0001_0037_4_00_00.dat
gla06_002_11_0001_0038_1_00_00.dat
gla06_002_11_0001_0038_2_00_00.dat
gla06_002_11_0001_0038_3_00_00.dat
gla06_002_11_0001_0038_4_00_00.dat
gla06_002_11_0001_0039_1_00_00.dat
gla06_002_11_0001_0039_2_00_00.dat
gla06_002_11_0001_0039_3_00_00.dat
gla06_002_11_0001_0039_4_00_00.dat
gla06_002_11_0001_0040_1_00_00.dat
gla06_002_11_0001_0040_2_00_00.dat
gla06_002_11_0001_0040_3_00_00.dat
gla06_002_11_0001_0040_4_00_00.dat
gla06_002_11_0001_0041_1_00_00.dat
gla06_002_11_0001_0041_2_00_00.dat
gla06_002_11_0001_0041_3_00_00.dat
gla06_002_11_0001_0041_4_00_00.dat
gla06_002_11_0001_0042_1_00_00.dat
gla06_002_11_0001_0042_2_00_00.dat
gla06_002_11_0001_0042_3_00_00.dat
gla06_002_11_0001_0042_4_00_00.dat
gla06_002_11_0001_0043_1_00_00.dat
gla06_002_11_0001_0043_2_00_00.dat
gla06_002_11_0001_0043_3_00_00.dat
gla06_002_11_0001_0043_4_00_00.dat
qap06_002_11_0001_0028_2_00_00.dat
qap06_002_11_0001_0028_3_00_00.dat
qap06_002_11_0001_0028_4_00_00.dat
qap06_002_11_0001_0029_1_00_00.dat
qap06_002_11_0001_0029_2_00_00.dat
qap06_002_11_0001_0029_3_00_00.dat
qap06_002_11_0001_0029_4_00_00.dat
qap06_002_11_0001_0030_1_00_00.dat
qap06_002_11_0001_0030_2_00_00.dat
qap06_002_11_0001_0030_3_00_00.dat
qap06_002_11_0001_0030_4_00_00.dat
qap06_002_11_0001_0031_1_00_00.dat
qap06_002_11_0001_0031_2_00_00.dat
qap06_002_11_0001_0031_3_00_00.dat
qap06_002_11_0001_0031_4_00_00.dat
qap06_002_11_0001_0032_1_00_00.dat
qap06_002_11_0001_0032_2_00_00.dat
qap06_002_11_0001_0032_3_00_00.dat
qap06_002_11_0001_0032_4_00_00.dat
```

```
qap06_002_11_0001_0033_1_00_00.dat
qap06_002_11_0001_0033_2_00_00.dat
qap06_002_11_0001_0033_3_00_00.dat
qap06_002_11_0001_0033_4_00_00.dat
qap06_002_11_0001_0034_1_00_00.dat
qap06_002_11_0001_0034_2_00_00.dat
qap06_002_11_0001_0034_3_00_00.dat
qap06_002_11_0001_0034_4_00_00.dat
qap06_002_11_0001_0035_1_00_00.dat
qap06_002_11_0001_0035_2_00_00.dat
qap06_002_11_0001_0035_3_00_00.dat
qap06_002_11_0001_0035_4_00_00.dat
qap06_002_11_0001_0036_1_00_00.dat
qap06_002_11_0001_0036_2_00_00.dat
qap06_002_11_0001_0036_3_00_00.dat
qap06_002_11_0001_0036_4_00_00.dat
qap06_002_11_0001_0037_1_00_00.dat
qap06_002_11_0001_0037_2_00_00.dat
qap06_002_11_0001_0037_3_00_00.dat
qap06_002_11_0001_0037_4_00_00.dat
qap06_002_11_0001_0038_1_00_00.dat
qap06_002_11_0001_0038_2_00_00.dat
qap06_002_11_0001_0038_3_00_00.dat
qap06_002_11_0001_0038_4_00_00.dat
qap06_002_11_0001_0039_1_00_00.dat
qap06_002_11_0001_0039_2_00_00.dat
qap06_002_11_0001_0039_3_00_00.dat
qap06_002_11_0001_0039_4_00_00.dat
qap06_002_11_0001_0040_1_00_00.dat
qap06_002_11_0001_0040_2_00_00.dat
qap06_002_11_0001_0040_3_00_00.dat
qap06_002_11_0001_0040_4_00_00.dat
qap06_002_11_0001_0041_1_00_00.dat
qap06_002_11_0001_0041_2_00_00.dat
qap06_002_11_0001_0041_3_00_00.dat
qap06_002_11_0001_0041_4_00_00.dat
qap06_002_11_0001_0042_1_00_00.dat
qap06_002_11_0001_0042_2_00_00.dat
qap06_002_11_0001_0042_3_00_00.dat
qap06_002_11_0001_0042_4_00_00.dat
qap06_002_11_0001_0043_1_00_00.dat
qap06_002_11_0001_0043_2_00_00.dat
qap06_002_11_0001_0043_3_00_00.dat
qap06_002_11_0001_0043_4_00_00.dat
gla12_002_11_0001_0015_0_00_00.dat
gla13_002_11_0001_0015_0_00_00.dat
gla14_002_11_0001_0015_0_00_00.dat
gla15_002_11_0001_0015_0_00_00.dat
gla12_002_11_0001_0029_0_00_00.dat
gla13_002_11_0001_0029_0_00_00.dat
gla14_002_11_0001_0029_0_00_00.dat
gla15_002_11_0001_0029_0_00_00.dat
gla12_002_11_0001_0043_0_00_00.dat
gla13_002_11_0001_0043_0_00_00.dat
gla14_002_11_0001_0043_0_00_00.dat
gla15_002_11_0001_0043_0_00_00.dat
qap12_002_11_0001_0015_0_00_00.dat
qap13_002_11_0001_0015_0_00_00.dat
qap14_002_11_0001_0015_0_00_00.dat
```

```
qap15_002_11_0001_0015_0_00_00.dat
qap12_002_11_0001_0029_0_00_00.dat
qap13_002_11_0001_0029_0_00_00.dat
qap14_002_11_0001_0029_0_00_00.dat
qap15_002_11_0001_0029_0_00_00.dat
qap12_002_11_0001_0043_0_00_00.dat
qap13_002_11_0001_0043_0_00_00.dat
qap14_002_11_0001_0043_0_00_00.dat
qap15_002_11_0001_0043_0_00_00.dat
```

#### 4.5.6.3 Procedure

In order to run this test individually, change to the $GLAS_HOME/test/test6 directory and run the test script.

```
cd $GLAS_HOME/test/test6
./run.sh
```

Examine the output to verify the results are successful and that the newly-created test data match the reference data.

Section 5

# Evaluation Criteria

## 5.1 Test 1: GLAS_L0proc

To pass acceptance testing, all tests performed must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

None.

## 5.2 Test 2: GLAS_L1A

To pass acceptance testing, all tests performed must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

None other than documented in the L1A science team product validation.

## 5.3 Test 3: Utility

To pass acceptance testing, all tests performed must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

Several tests were skipped, as documented in the Utility Integration Test Report.

## 5.4 Test 4: GLAS_Atm

To pass acceptance testing, all tests performed must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

GLA10 and GLA11 creation is not tested.
None other than documented in the Atmosphere science team product validation

## 5.5 Test 5: GLAS_Alt - Waveform

To pass acceptance testing, all tests performed must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

None other than documented in the Waveforms science team product validation.

## 5.6 Test 6: GLAS_Alt - Elevation

To pass acceptance testing, all tests performed must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

None other than documented in the Elevations science team product validation.

Section 6

# Expected Results

## 6.1 Test 1: GLAS_L0proc

Output of a successful acceptance test is shown below:

```
Verifying Test 1 outputs....


Passed: gla01_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla01_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla01_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla02_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla02_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla02_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla03_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla03_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla03_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

## 6.2 Test 2: GLAS_L1A

Output of a successful acceptance test is shown below:

```
Verifying Test 2 outputs....


Passed: gla05_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla05_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla05_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

## 6.3 Test 3: Utility

Output of a successful acceptance test is shown below:

```
Verifying Test 3 outputs....
```

```
Passed: gla07_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla07_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla07_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla08_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla08_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla08_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla09_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla09_001_11_001_0001_2_01_00.dat matches the reference.

Passed: gla09_001_11_001_0001_3_01_00.dat matches the reference.

Passed: gla10_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla10_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla10_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla11_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla11_001_11_001_0001_2_01_00.dat matches the reference.

Passed: gla11_001_11_001_0001_3_01_00.dat matches the reference.

Done.
```

## 6.4 Test 4: GLAS_Atm

Output of a successful acceptance test is shown below:

```
Verifying Test 4 outputs....


Passed: gla06_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla12_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla12_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla12_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla13_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla13_001_11_001_0001_2_01_00.dat matches the reference
```

```
Passed: gla13_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla14_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla14_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla14_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla15_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla15_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla15_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

## 6.5 Test 5: GLAS_Alt - Waveforms

Output of a successful acceptance test is shown below:

```
Verifying Test 5 outputs....

Passed: gla06_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

## 6.6 Test 6: GLAS_Alt - Elevations

Output of a successful acceptance test is shown below:

```
Verifying Test 5 outputs....

Passed: gla06_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

Section 7

# Actual Results

## 7.1 Test 1 : GLAS_L0proc

Passed. Acceptance test outputs exactly matched the integration test references.

## 7.2 Test 2 : GLAS_L1A

Passed. Acceptance test outputs exactly matched the integration test references.

## 7.3 Test 3 : Utility

Passed. Acceptance test outputs exactly matched the integration test references.

## 7.4 Test 4: GLAS_Atm

Passed. Acceptance test outputs exactly matched the integration test references.

## 7.5 Test 5 : GLAS_Alt - Waveform

Passed. Acceptance test outputs exactly matched the integration test references.

## 7.6 Test 6 : GLAS_Alt - Elevation

Passed. Acceptance test outputs exactly matched the integration test references.

# Abbreviations & Acronyms

| | |
|---|---|
| ALT | designation for the EOS-Altimeter spacecraft series |
| DAAC | Distributed Active Archive Center |
| EDOS | EOS Data and Operations System |
| EOC | EOS Operating Center |
| EOS | NASA Earth Observing System Mission Program |
| EOSDIS | Earth Observing System Data and Information System |
| GDS | GLAS Ground Data System |
| GLAS | Geoscience Laser Altimeter System instrument or investigation |
| GPS | Global Positioning System |
| GSFC | NASA Goddard Space Flight Center at Greenbelt, Maryland |
| GSFC/WFF | NASA Goddard Space Flight Center/Wallops Flight Facility at Wallops Island, Virginia |
| ID | Identification |
| IEEE | Institute for Electronics and Electrical Engineering |
| IST | GLAS Instrument Support Terminal |
| LASER | Light Amplification by Stimulated Emission of Radiation |
| LIDAR | Light Detection and Ranging |
| N/A | Not (/) Applicable |
| NASA | National Aeronautics and Space Administration |
| NOAA | National Oceanic and Atmospheric Administration |
| POD | Precision Orbit Determination |
| SCF | GLAS investigation Science Computing Facility and workstation(s) |
| SDPS | Science Data Processing Segment |
| TBD | to be determined, to be done, or to be developed |
| UNIX | the operating system jointly developed by the AT&T Bell Laboratories and the University of California-Berkeley System Division |

# Glossary

| | |
|---|---|
| aggregate | A collection, assemblage, or grouping of distinct data parts together to make a whole. It is generally used to indicate the grouping of GLAS data items, arrays, elements, and EOS parameters into a data record. For example, the collection of Level 1B EOS Data Parameters gathered to form a one-second Level 1B data record. It could be used to represent groupings of various GLAS data entities such as data items aggregated as an array, data items and arrays aggregated into a GLAS Data Element, GLAS Data Elements aggregated as an EOS Data Parameter, or EOS Data Parameters aggregated into a Data Product record. |
| array | An ordered arrangement of homogenous data items that may either be synchronous or asynchronous. An array of data items usually implies the ability to access individual data items or members of the array by an index. An array of GLAS data items might represent the three coordinates of a georeference location, a collection of values at a rate, or a collection of values describing an altimeter waveform. |
| file | A collection of data stored as records and terminated by a physical or logical end-of-file (EOF) marker. The term usually applies to the collection within a storage device or storage media such as a disk file or a tape file. Loosely employed it is used to indicate a collection of GLAS data records without a standard label. For the Level 1A Data Product, the file would constitute the collection of one-second Level 1A data records generated in the SDPS working storage for a single pass. |
| header | A text and/or binary label or information record, record set, or block, prefacing a data record, record set, or a file. A header usually contains identifying or descriptive information, and may sometimes be embedded within a record rather than attached as a prefix. |
| item | Specifically, a data item. A discrete, non-decomposable unit of data, usually a single word or value in a data record, or a single value from a data array. The representation of a single GLAS data value within a data array or a GLAS Data Element. |
| label | The text and/or binary information records, record set, block, header, or headers prefacing a data file or linked to a data file sufficient to form a labeled data product. A standard label may imply a standard data product. A label may consist of a single header as well as multiple headers and markers depending on the defining authority. |
| Level 0 | The level designation applied to an EOS data product that consists of raw instrument data, recorded at the original resolution, in time order, with any duplicate or redundant data packets removed. |
| Level 1A | The level designation applied to an EOS data product that consists of reconstructed, unprocessed Level 0 instrument data, recorded at the full resolution with time referenced data records, in time order. The data are annotated with ancillary information including radiometric and geometric calibration coefficients, and georeferencing parameter data (i.e., ephemeris data). The included, computed coefficients and parameter data have not however been applied to correct the Level 0 instrument data contents. |

| | |
|---|---|
| Level 1B | The level designation applied to an EOS data product that consists of Level 1A data that have been radiometrically corrected, processed from raw data into sensor data units, and have been geolocated according to applied georeferencing data. |
| Level 2 | The level designation applied to an EOS data product that consists of derived geophysical data values, recorded at the same resolution, time order, and georeference location as the Level 1A or Level 1B data. |
| Level 3 | The level designation applied to an EOS data product that consists of geophysical data values derived from Level 1 or Level 2 data, recorded at a temporally or spatially resampled resolution. |
| Level 4 | The level designation applied to an EOS data product that consists of data from modeled output or resultant analysis of lower level data that are not directly derived by the GLAS instrument and supplemental sensors. |
| metadata | The textual information supplied as supplemental, descriptive information to a data product. It may consist of fixed or variable length records of ASCII data describing files, records, parameters, elements, items, formats, etc., that may serve as catalog, data base, keyword/value, header, or label data. This data may be parsable and searchable by some tool or utility program. |
| orbit | The passage of time and spacecraft travel signifying a complete journey around a celestial or terrestrial body. For GLAS and the EOS ALT-L spacecraft each orbit starts at the time when the spacecraft is on the equator traveling toward the North Pole, continues through the equator crossing as the spacecraft ground track moves toward the South Pole, and terminates when the spacecraft has reached the equator moving northward from the South Polar region. |
| model | A graphical representation of a system. |
| module | A collection of program statements with four basic attributes: input and output, function, mechanics and internal data. |
| parameter | Specifically, an EOS Data Parameter. This is a defining, controlling, or constraining data unit associated with a EOS science community approved algorithm. It is identified by an EOS Parameter Number and Parameter Name. An EOS Data Parameter within the GLAS Data Product is composed of one or more GLAS Data Elements |
| pass | A sub-segment of an orbit, it may consist of the ascending or descending portion of an orbit (e.g., a descending pass would consist of the ground track segment beginning with the northernmost point of travel through the following southernmost point of travel), or the segment above or below the equator; for GLAS the pass is identified as either the northern or southern hemisphere portion of the ground track on any orbit |
| PDL | Program Design Language (Pseudocode). A language tool used for module programming and specification. It is at a higher level than any existing compilable language. |
| process | An activity on a dataflow diagram that transforms input data flow(s) into output data flow(s). |

| | |
|---|---|
| product | Specifically, the Data Product or the EOS Data Product. This is implicitly the labeled data product or the data product as produced by software on the SDPS or SCF. A GLAS data product refers to the data file or record collection either prefaced with a product label or standard formatted data label or linked to a product label or standard formatted data label file. Loosely used, it may indicate a single pass file aggregation, or the entire set of product files contained in a data repository. |
| program | The smallest set of computer instructions that can be executed as a stand-alone unit |
| record | A specific organization or aggregate of data items. It represents the collection of EOS Data Parameters within a given time interval, such as a one-second data record. It is the first level decomposition of a product file. |
| Scenario | A single execution path for a process. |
| Standard Data Product | Specifically, a GLAS Standard Data Product. It represents an EOS ALT-L/GLAS Data Product produced on the EOSDIS SDPS for GLAS data product generation or within the GLAS Science Computing Facility using EOS science community approved algorithms. It is routinely produced and is intended to be archived in the EOSDIS data repository for EOS user community-wide access and retrieval. |
| State Transition Diagram | Graphical representation of one or more scenarios. |
| Stub | (alias dummy module) a primitive implementation of a subordinate module, which is normally used in the top-down testing of superordinate (higher) modules. |
| Structured Chart | A graphical tool for depicting the partitioning of a system into modules, the hierarchy and organization of those modules, and the communication interfaces between the modules. |
| Structured Design | The development of a blueprint of a computer system solution to a problem, having the same components and interrelationships amount the components as the original problem has. |
| Subroutine | A program that is called by another program |
| variable | Usually a reference in a computer program to a storage location, i.e., a place to contain or hold the value of a data item. |